

开源时代

OPEN SOURCE TIMES

2009年8月刊 总第十一期

社区扫描:

Linux基金会推Visa白金信用卡促开源发展
管理员最终出现 CentOS项目危机化解

开源业界:

USPS包裹跟踪系统平台迁移到GNU/Linux
红帽跻身标普500指数 成该指数第一家Linux企业

技术新知:

软RAID升级—单盘分身大法
独辟蹊径之Linux主机监控工具
munin monit ntop篇

行业观察:

评论: 微软开源姿
态的真正含义
外电评论: 开源在
中国面临重重困难

ChinaUnix专访:

Linux基金会主席Jim Zemlin
Ubuntu Linux创始人Mark Shuttleworth
OpenOffice.org社区开发经理Louis Suárez-Vázquez

2009' 开源中国 开源世界 高峰论坛
2009' Open Source China Open Source World Summit

圆桌会议



本期推荐: 生于有效创新, 死于盲目仿制



ChinaUnix.net



2009年8月刊 总第十一期

编辑出品:  ChinaUnix.net
网络发行: ChinaUnix

主编: 江晖
技术主编: 樊强
执行编辑: 周荣茂
内容编辑: 周荣茂 覃里
唐川 周平
技术编委: 高延斌 马路遥
白金
美术编辑: 林在子
交流论坛: bbs.chinaunix.net
杂志下载: www.chinaunix.net
www.itpub.net
www.ixpub.net

联系我们:
rmzhou@staff.chinaunix.net
投稿邮箱:
rmzhou@staff.chinaunix.net

媒体支持:  IT168.com

广告联系: 温玉琴
电话: 010-82658790
手机: 13801339139
E-mail: wyq@it168.com

内容目录

卷首语

004

开源业界

美国邮政总局包裹跟踪系统平台迁移到GNU/Linux	005
红帽跻身标普500指数 成该指数第一家Linux企业	005
开源对智能手机有重要意义的10大理由	006
Linux内核新补丁: 巧妙规避微软文件系统专利	008
Oracle手中另一张对抗微软的王牌: Ubuntu	009
Zend创始人访谈: 不急于升级到PHP6	010
Mozilla志愿者开发模式被复制	012
开源协议GPL份额下降 GPLv3增长	013
诺基亚欲使用Qt技术构造Maemo/塞班一体化开发平	014
印度政府欲自主研发处理器	015
微软向开源示好: 发布2万行Linux设备驱动代码	015
Intel成第二大Linux内核代码贡献者	016
Linux最新0day的幕后故事和技术分析	017
云操作系统让思科与Red Hat亦敌亦友	018

社区扫描

Linux基金会推Visa白金信用卡促开源发展	020
不堪linus批评 Alan Cox选择退出	020
管理员最终出现 CentOS项目危机化解	021
SourceForge2009社区选择奖名单公布	022
Debian采用两年的发行周期	022
Firefox 3.5的启动堪比操作系统	023
Ubuntu积极推动开源发展	023
巴西总统头戴ODF帽现身自由软件会议	024
Oracle可能将终止OpenSolaris项目	024

Ubuntu：下一版本不一定采用GNOME3.0	024
最新Linux Kernel发现0day安全漏洞	025
开源社区协作工具Launchpad源码公布	025
Oracle为什么不会干掉MySQL	026

行业观察

甲骨文需要面对的一个开源竞争对手：Ingres	027
开源技术依然存在不足之处需要改进	028
谷歌Android进军企业市场之路漫长而曲折	029
外电评论：开源在中国面临重重困难	031
谷歌的开源：窃取胜利果实，还是拥抱社群？	031
评论：微软开源姿态的真正含义	033
传统路由技术的中年危机和开源新势力	034

专家专栏

国内开源尚需耕耘 国际交流凸显重要	037
-------------------	-----

本期推荐

生于有效创新 死于盲目仿制	042
---------------	-----

技术新知

独辟蹊径之Linux主机监控工具munin monit ntop	046
软Raid升级—单盘分身大法	061
带你剖析Linux目录结构	070
不输入密码自动通过SSH方式登录服务器	072
RHEL5实现笔记本电脑触摸板开关功能	074
Linux下patch的制作和应用	076
使用qemu进行内核源码级调试	084
Rootkit技术的主要原理	088

网友热评	091
------	-----

版权声明

杂志内容来自ChinaUnix社区及互联网，电子杂志的宗旨是为了更好地传递开源最新自寻和技术经验。如有版权问题敬请联系，我们将会第一时间做出处理。

致谢

本杂志得到ChinaUnix网站Linux时代社区版主的大力支持，技术文章大部分来自版主推荐，更多技术文章可以访问Linux时代精华区。
本刊分析评论部分文章来自IT168技术频道。

卷首语

网络是虚拟的，那么人与人之间就会少了一份敬重。上个月在开源中国开源世界高峰论坛会，我们采访 OpenOffice.org 社区开发经理 Louis Suarez-Potts 的时候，他如是地说。这个月来发生在开源技术社区的事情就验证了这一点。先是 Linus 无情地批评了 Alan Cox 作为内核 TTY 子项目负责人无所作为，然后 Alan Cox 发表声明，宣称将不再担任该项目负责人。Alan Cox 现在为 Intel 公司工作，之前在 Red Hat 公司供职长达 10 年，是 Linux 和自由软件社区的资深开发人员，从 1991 年开始，就为 Linux 内核贡献代码；Linus 当然不用介绍，作为 Linux 的创始人，在开源社区的地位当然不容置疑。为何两人之间会爆发如此大的分歧？社区会员众说纷纭，各方拥趸悉数登场。有的说，Alan Cox 在加入 Intel 之后，将会更多的考虑到公司的利益，不得将工作重心有所转移；而 Linus 的反对者则认为，Linus 现在有脱离自由社区精神的原则，和开发人员和贡献者的关系越搞越差。

此事还没有过去几天，开源社区又出了另外一件大事，CentOS Linux 项目管理员神秘失踪，一则项目管理团队的告社区用户书，一下子把 CentOS 用户社区搞得哗然一片。但是没有过几天，项目管理团队又重新找到了这位管理人员，并且就相关问题达成了一致。也许是虚惊一场，但是这折射出的开源社区的管理问题，我们不得不重视，正如一位网友的评论：“开源项目没那么容易做，人多嘴杂，组织也很松散。更重要的是，不像商业炒作，没有人从中获利，大家的动力仅仅是爱好，没有了热情，项目必将走向消亡。”难道开源项目的最终目标是如此么？确实，CentOS 事件反映出的不仅仅是个别社区存在的问题，更多的是开源项目社区普遍存在的问题。这也许能够说明，开源社区不仅仅是技术社区，不只是需要技术开发人员，还需要项目管理人员，而且是有经验的项目管理人员，否则项目管理这块将是开源项目走向成功的致命短板。这也许是一个开源项目和一个公司里的项目所存在的本质区别。

过去的几周，全球范围内一系列 Linux 和开源软件产业相关的展会相继上演。在渥太华举行的 Linux 峰会上，LWN.net 创办人 Jonathan Corbet 公布了有关 Linux 内核代码开发者的统计数据，Red Hat 连续三年成为内核代码主要开发厂商，而 Intel 则首次成为第二大代内核代码开发厂商，主要是因为 Intel 正大力推动 Moblin 操作系统，这同时也代表着 Intel 看好未来 Linux 操作系统前景。我们看到 Intel 公司正在为 Linux 内核做出越来越多的贡献，也许是商业利益驱使，但是我们还是看到 Wintel 联盟最终走向灭亡的趋势。紧接着，7 月 20 日-24 日，Oreilly 组织的 OSCON2009 在美国加州 San Jose 举办。作为传统开源领域的技术和商业创新会议，吸引了超过一万多名技术用户和开源爱好者，超过八十多家开源相关技术企业参展和参与大会讨论。随着开源理念的深入人心，全球重要的开源技术和商业会议，微软现在基本全勤，这次也不例外。并且在会上宣布为 Linux 内核贡献超过两万行的虚拟化技术相关驱动代码，此外，微软也将为教育行业开源软件 Moodle 带来新的服务器插件，该插件程序能使 Moodle 获得微软 Live Services，包括 email、即时通信、日历等。

最后，再次提醒大家，CU 将联合 IT168 在 8 月 28 日举办中国系统架构师大会，各项准备工作有条不紊地进行当中，最新大会进展请关注大会官网和 ChinaUnix 网站的更新。

ChinaUnix 社区编辑：周荣茂

开源业界

美国邮政总局包裹跟踪系统平台迁移到 GNU/Linux

美国邮政总局日前将他们的包裹跟踪系统的运行环境更换到 Linux 平台。以前的这个跟踪系统是由 Cobol 语言编写的，目前已经为 Linux 平台做过相关转换，且可以在 Linux 平台良好运行，这可比将以前的系统重写一遍要来的便宜。



当前经济危机下，各国的邮政和快递物流业务日趋萧条，特别是在美国，有 UPS 和 Fedex 等强有力的竞争对手，USPS 面临着非常大的生存压力，削减成本算成为其一个有力的竞争手段。包裹跟踪系统运行平台的迁移，是美国邮政总局采用开源软件降低运行成本、提高系统交易处理能力计划的一部分。目前，USPS 每天处理大约四千万份交易，为美国邮局系统提供从挂号信到邮政包裹的跟踪服务。

尽管 USPS 相关负责人并没有透露详细的数字，但是显而易见，从 Solaris 的软硬件转移到 HP 的硬件和 Linux 操作系统，肯定省下了一大笔钱。邮政系统的代表 John Byrne 告诉记者，虽然省下很多，但是不像从 IBM 的专有环境迁移过来，因为 IBM 的专有硬件设备太贵，所以省下的并不是大家想像的那么多。

在目前的经济不景气的大潮下，我们很高兴看到 USPS 能够通过采用开源软件的方式来度过时艰。虽然现在在美国政府内部还是存在着对开源软件的反对意见，FOSS 还是在埋头奋进，或许我们就在成功的路上。

红帽跻身标普 500 指数 成该指数第一家 Linux 企业

据国外媒体报道，金融危机的不断蔓延，造成整个华尔街损失的日益增加，但是对于红帽公司而言，金融危机却带来了使其跻身标准普尔 500 指数成份股的机会，而该公司在整个企业界的地位也将进一步上升。

根据计划，红帽将从下周开始取代深陷破产困境的金融巨头 CIT 公司，正式加入标准普尔 500

指数，成为标准普尔 500 指数中的第一家 Linux 企业。对于红帽而言，加入标普 500 指数所带来的不仅仅是表面上的荣誉，更为实际的是，该公司股票将受到那些一直以标普 500 指数为参照的投资者的追捧。

受到外界对公司前景持乐观态度的影响，红帽公司股价在 2009 年迄今的累计涨幅已经超过了 50%。投资者认为，消费者在经济危机的情况下将选择相对廉价的产品，而红帽公司所提供的 Linux 相关软件产品无疑迎合了市场的这种需求。

截至当日收盘，红帽股价下跌了 26 美分，报每股 20.60 美元。但是随后标准普尔公司宣布将其纳入标准普尔 500 指数，该公司股价在当天的盘后交易中大涨了 10%。



根据计划，红帽将从 7 月 24 日美国股市收盘之后正式加入标准普尔 500 指数。由于金融危机的影响，一些大公司纷纷陷入破产困境，为了保持其指数的含金量，标准普尔一直在对标普 500 指数的成份股进行调整。

自 2009 年以来，CIT 公司的股价已经累计下跌了 85%。由于受到金融危机的巨大冲击，投资者对于这家公司的前景充满担忧。CIT 是一家小企业短期贷款机构，并且主要为零售行业的公司提供贷款。

开源对智能手机有重要意义的 10 大理由

开源为手机市场带来了太多的好处，除了节约成本外还有其它很多优势，从更安全到更多定制选项再到更多的应用程序开发。

手机产业是很有趣的，我们终于看到智能手机开始真正变得智能，智能手机用户从其获得的好处变得越来越多，这是怎么发生的呢？用一个单词概括就是：竞争！

当苹果的 iPhone 一上市就遭到用户的疯狂追捧，其竞争对手开始进行不停地模仿，竞争在一直不断加剧，Android 手机、Palm Pre 和黑莓这些都早已加入到战斗中来了，但我认为它们之中只有两个会最终胜出，原因只有一个，那就是开源，那么为什么开源会帮助这些手机提升竞争力呢？我认为有 10 大原因。

1、开放的标准

选择 iPhone 后你就得跟着苹果的标准走，你就只能使用苹果认可的应用程序(除非你对你的 iPhone 进行越狱)，如果选择基于 Android 的手机和 Palm pre，它们可不是说开放一点点内容或一段时间，它们将一直坚持走开放的标准。软件将更容易开发，网站负荷将符合预期(也更容易为移动设备开发)，硬件配件将更容易获得。

2、更多的应用程序

毕竟，苹果是应用程序的王国，似乎你想到的任何东西苹果都有一个对应的应用程序，但随着 Android 和 pre 更广泛的使用，对应的应用程序数量将呈指数级增加，为什么呢？首先，应用程序开发不受到苹果那样的认可过程，任何时候你可以开发任何应用程序，苹果就不一样了，当你要开发的东西已经内置到 iPhone 中的话，那等待你的将会是一个折磨人的过程。你想在 iPhone 上使用不同的浏览器，对不起，你没得选择。我找到了 Firefox 和 chrome 为 Android 和 pre 发布的移动版本，这种情况还是持续下去，知道其中一个(或两个都)应用程序商店超过了苹果的应用程序商店。

3、安全更好

安全迟早会成为移动计算的重大问题，苹果为放慢 iPhone 升级补丁已经显得很痛苦了，由于开源的本身特性，升级就不会显得这么慢了，因此当发现一个安全漏洞或缺陷时，用户将能够更快获得升级补丁，当然，这不仅仅只针对升级补丁，pre 和 Android 都是基于 Linux 的，因此它们的安全基础要好一些，虽然手机安全问题还没有大量出现，随着智能手机的普及，安全问题将很快会引起重视。

4、可量身定制

我有一部一代 iPhone 手机，我最大的抱怨是它不能完全按我的意愿进行定制，它没有主题，我个人认为这是一个巨大的失败，在开源智能手机中，你可以变换主题，并可以自己定做自己的主题，过半数用户希望他/她们能够按照自己想要的设计外观，这会让智能手机工作得更好吗？当然不会，但这个功能会激发用户的兴趣。

5、连通性更好

我这里不是说 3G、EDGE 和 Wi-Fi，我说的是连接到 PC。使用 iPhone 时你可以用 iTunes 进行同步，如果你注册了 MobileMe，你就可以用一个迂回的办法来同步 Gmail 了，Pre 会在你的计算机中显示为一个标准的大容量存储设备，因此可以进行拖拉操作。同步几乎可以在所有平台上运行，因此使用 Pre 和 Android 手机时，可以和 MAC、Windows 和 Linux 进行同步，实现了真正的跨平台，这一点大大超出了 Windows 和苹果手机。

6、成本更低

我已经计划转移到 Android 或 Pre 了，吸引我的一个重要原因就是成本，如果捆绑 2 年服务的 iPhone 3G，我将花去 3799 美元，而 Android G1 是 3149 美元，Pre 是 2599 美元，平均每个月要便宜 22 美元，这样计算我还可以多买一部智能手机了，它们是如何保持这种低成本的呢？答案就是操作系统是不要钱的，开源的。

7、支持多任务

这是另一个 iPhone 让我烦恼的事情，如果我在 EDGE 网络上，不小心点击了邮件按钮，我可以使用电话的通话功能进行放弃，但我曾经因此切断了很多呼入电话，iPhone 还不能支持多任务，但 Android 和 Pre 都可以支持，你可以一次打开多个程序，如果你在 Pre 上，你会感觉非常自由，如果你 iPhone 上，忘掉它吧，让我们面对现实吧，我们身处一个多任务的社会，为什么你还想使

用智能手机操作系统中的 DOS 呢？Android 和 Pre 都是基于 Linux 创建的，它们都是支持多任务和网络的。

8、Gmail 推送

大部分 Google 应用程序都内置到 Pre OS 中了，其中 Gmail 也可以直接推送到你的 Pre 手机中，你甚至不用打开邮件客户端邮件就可以进行下载，当你打开邮件客户端时邮件已经躺在那儿了，这个功能也出现在 Android 手机上，当然，你可以让你的 iPhone 频繁地执行邮件检查，让它看起来就象是推送一样(调侃ing)。Google 已经发布了一个开源 Gmail API，Pre 和 Android 与 Gmail 的集成度将会更好。

9、开发人员众多

现在已经有大量的开发人员加入到 Android 和 Pre 的开发网络中来，全球开源开发人员的数量是相当令人惊讶的，想象一下集体的智慧吧，通过多人协作努力开发出的应用程序更加符合人们的需要。这种模式在 Linux 操作系统身上已经被证明是非常有效的，当发现一个漏洞时，很快就会得到修复，Android 和 Pre 亦如此，当很多人为同一个目标努力工作时，那么这个目标是不是很快就会实现呢？

10、更有创造力

你认为开源社区需要多长时间开发一个 Apache 超轻量级版本运行在 Pre 呢？这样你就可以随身携带你的网站了，不仅如此，开源社区还会发现许多创新性的东西，如邮件服务器，CMS，网络安全工具等等，真是要印证那句话“一切皆有可能！”，可能过不了多久，你将会在移动设备上看到 Linux 桌面了。

小结

成本固然是开源带来的巨大优势，现在你应该看看如何加入到开源社区为智能手机做点贡献。你能想到其它开源对智能手机的好处吗？你能想到有什么方法阻碍它们吗？欢迎发表你的意见！

Linux 内核新补丁：巧妙规避微软文件系统专利

微软最近控告 TomTom 公司侵犯了自己的 FAT 文件系统专利，这个事件在开源界尤其是使用 Linux 操作系统和 FAT 文件系统的企业用户造成了不小的影响，而上周有关人员为此专门发布了 Linux 操作系统的新内核补丁，由于该补丁对微软的专利进行了规避，因此安装这款补丁后，Linux 操作系统的用户就可以继续使用操作系统操作 FAT 文件系统而不会受到微软的控告。

TomTom 是一家专门生产便携式导航设备的厂家，他们部分 GPS 产品中安装了基于 FAT 文件系统的 Linux 操作系统。而今年二月份，微软则对这家公司提起了诉讼，他们宣称 TomTom 的这些 GPS 产品侵犯了自己的 FAT 文件系统专利。目前这种文件系统在便携存储设备中应用得非常广泛，USB 闪存盘，闪存卡等产品都经常使用这种文件系统。

不甘示弱的 TomTom 随后发起了反诉，不过三月份终审判决过后 TomTom 败诉，他们被迫在自己的产品中移除了与 FAT 文件系统有关的内容。

不过，这次判决的结果却令使用 FAT 文件系统的 Linux 操作系统的合法性变得非常含糊不清。虽然微软宣称不会对使用 FAT 文件系统的 Linux 个人用户进行起诉，但这对那些使用 FAT 文件系统和 Linux 操作系统的企业而言则非常头疼。目前使用 Linux 操作系统的移动设备不在少数，而这些设备都需要具备能读取 FAT 格式存储体的功能。

Linux 基金会组织宣称最好的方法就是放弃 FAT 文件系统，并重新开发一套新的文件格式。长远上看，这个提议当然是非常合理的，不过眼下 Linux 系统还很难做到马上就放弃对 FAT 格式的支持，而新补丁的推出则正好可以解决这个问题。

微软 FAT 专利问题主要集中在一份名为《长短文件名通用名字空间》的专利上。早期的 MsDOS 操作系统只支持 11 个字符长度的文件名，即所谓的 8.3 命名机制。为了满足向下兼容性的要求，这份专利同时为文件分配一个短文件名记录项和多个长文件名记录项。

而上周发布的 Linux 补丁则改变了 Linux 操作系统处理 FAT 文件系统的方法，现在 Linux 不会为同一个文件同时分配长/短文件名记录。如果文件名字符长度没有超过 11 个字符，那么系统将只为这个文件分配一个短文件名记录项；而如果文件名超过了 11 个字符，系统将只为这个文件分配一个长文件名记录项，并在原来的短文件名记录项中填上 11 个非法字符，这样长文件名文件的短文件名记录项就会被操作系统视为非法文件名而不会显示出来。

补丁的作者解释说：微软的 VFAT 文件系统有关的两份专利中采用的是为同一个文件同时分配长/短文件名记录项的方法，而我们的补丁则只会为同一个文件分配一种可用的文件名记录项。

打补丁后，原来的短文件名记录项将使用非法字符进行填充，这些非法字符将是随机生成的。而不是简单地用 0 值来进行填充，因为在这些记录项内填入特定的数字可能会引起 Windows XP 系统死机，而随机生成的非法字符则可以尽量避免出现这种情况。

这已经是补丁作者第二次发布于 FAT 专利事件有关的补丁了，在作者第一次发布的补丁中，他完全废除了长文件名，而新版补丁则由于保留了长文件名因此显得更加实用一些，也因此更受用户的欢迎。

尽管如此，Linux 基金会仍然坚信微软的所谓 FAT 专利是不合法的，而他们推出这些补丁的目的只是想让一些企业 Linux 用户尽量不会因此而遭到微软的骚扰，甚至被后者拖入麻烦的官司漩涡之中。

Oracle 手中另一张对抗微软的王牌：Ubuntu

Oracle 早就有意进军操作系统市场，来完善自己的产品线，其目的是同微软进行竞争。在自主研发还是收购现有 Linux 操作系统的问题上，Oracle 曾表态将收购一家现有的公司，被业内认为最有价值厂商的 Redhat 无疑是最佳人选。不过近日，一位接近 Oracle 的人士向我们解释了 Oracle 为什么不会收购 Red hat 的原因，根据他所掌握的数据，Oracle 公司在 Linux 操作系统平台下赢得了 80% 以上的市场份额，而在 Windows 平台下只有 50% 的占有率，而且这个比率由于微软 SQL Server 近期的优秀表现还在继续下降。因此，Oracle 当前的首要任务并不是肃清 Linux 平台上的 Red hat，而是要对付微软所造成的威胁，而他们手上现在也已经有了另一张王牌。

显然，短期内 Red hat 不会降低红帽子企业版的卖身价，至少对 Oracle 是这样。不过 Oracle 手中现在握有另一款颇具流行潜质，而且软件品质也颇高的 Linux 王牌：Ubuntu。

目前 Ubuntu 唯一的短板就是缺乏企业级大型软件厂商的支持，而 Oracle 则正好可以填补这一空缺，他们只要把 Ubuntu 加入自己的认证范围就可以达到这一目的。

有此王牌在手，Oracle 当然就不会考虑花费巨资来收购 Redhat，相比之下 Oracle 在 Ubuntu 上也许不花一分钱就可以与其厂商 Canonical 结成合作伙伴关系，而且眼下 Ubuntu 在企业和个人用户间也已经具备相当的流行度。

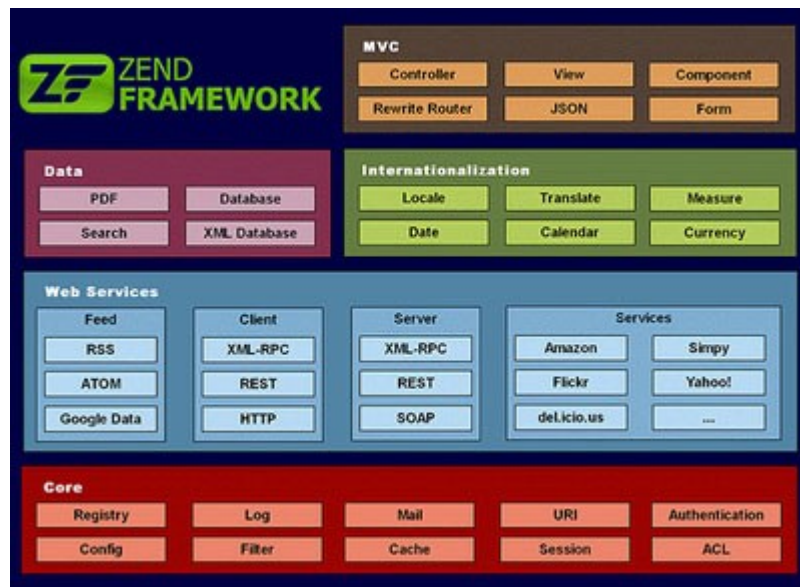
如果 Oracle 要想打败微软，就需要改变自己的产品推广模式，尽量与 Windows 的模式接近。

而要彻底改变自己产品的推广模式，则需要借用 Ubuntu 的力量。

Zend 创始人访谈：不急于升级到 PHP6

Zend 是众多开源成功故事之一的主角，它遵循了已在 MySQL 和 Subversion 中被证明成功的模式，针对那些需要高可靠性、高可扩展性 Web 应用服务器的企业客户，提供技术支持和商用软件。

此前，该公司已经宣布推出了 Zend Server 软件，这是一个在一台到三台服务器上运行 PHP 应用的高度集成平台。它包含一个高速代码缓存功能，可以让 PHP 应用更快速运行。它还可以监控用户服务器上的所有 PHP 脚本，当运行缓慢或出错时向用户告警。



现在，Zend 正在推出其企业软件的精简免费版，在 Zend Server 社区版中，用户可以获得同样的高性能代码执行和管理界面，但是不会获得技术支持，而且当 PHP 更新发布时，不会获得即时升级。尽管如此，用户至少可以使用其强大的功能，轻松节省昂贵的硬件升级成本。

Zend 创始人济夫*苏拉斯基(Zeev Suraski)最近接受了媒体采访，谈论了 Zend Server 社区版与其非免费版本的关系，以及其团队为何花费了如此长时间来开发 PHP6 等。

记者：Zend Server 与现有 Zend 平台软件有何异同？

苏拉斯基：开始阶段，我们计划同时运作管理这两款产品，其中 Zend Server 针对单服务器解决方案，最多两、三台服务器；而 Zend 平台针对大型集群环境。我们决定逐渐推出 Zend Server 战略，因此未来我们计划在 Zend Server 上增加群集支持功能，以及对其增加 Zend 平台上的其它功能和新功能。

记者：也就是说，你的计划是把 Zend 平台的功能加入到 Zend Server 中？

苏拉斯基：对，不过不仅仅增加来自 Zend 平台的功能，我们还将改进其用户界面和易用性，因此未来在群集版 Zend Server 中你不会看到完全相同的功能，而是它们的改进版；或者具有更高的性能，或者是全新的功能。

记者：Zend Server 的关键特点之一是，它是一个高度集成的解决方案，具有以更优化方式配置运行的软件。但用户已经有了 MySQL 和 Apache，是否真的还需要 Zend Server？

苏拉斯基：这要看你使用什么操作系统。如果你在一个已经安装 Apache 的 Linux 系统上安装它，那么你无需安装一个新 Apache，只需使用系统自带的 Apache 即可。但是如果你使用其它系统，你可以通过 Zend Server 来安装 Apache，这将非常方便。

记者：用户能使用系统自带的 PHP 吗？

苏拉斯基：不能。用户必须使用 Zend Server 中的 PHP。从技术的角度讲，你或许可以这么做，但是我们不建议这么做，也不对此进行支持。

记者：Zend Server 的监控代理自动跟踪脚本、数据库连接或其它 PHP 相关问题出错的时间，它能够保存应用程序状态，因此可以在以后进行恢复。这需要占用很多资源吗？

苏拉斯基：如果用户的应用程序确实非常差，那么它会产生很多这种事件，保存它们的确需要花费大量时间。但一般情况下，用户的应用程序不会产生太多这种事件，其资源开销几乎可以忽略不计，大约在 1%到 2%之间。它取决于产生事件的数量。

记者：是否可以说 Zend 正在进入以前由 Java 应用服务器统治的一个领域？

苏拉斯基：在某种程度上我认为是如此。实际上在某些方面已经是一个事实。PHP 已经被应用于许多关键业务应用中，并且被进行了非常大范围的部署，例如维基百科、YouTube 和 Flickr 等。这种趋势还在增长，我们认为这非常有意义，并在 Zend Server 中对其进行支持。

记者：另一方面，随着 Zend Server 让 PHP 更加面向企业，是否可以说 PHP 可能在失去与其最初社区的联系？

苏拉斯基：我认为不是这样，但是我要指出的是 PHP 已经发展了 12 年到 13 年，它不再是新手语言。也就是说，我认为 PHP 社区依然在开发它，而且与使用它的社区依然保持紧密联系。PHP 社区非常健康，它非常强大且依然在增长。

PHP 的关键优势是它是一个成熟的解决方案，这已经被事实证明。知道如何使用 Ruby 或 Python 部署网站的人还相对较少，它们也都是非常好的解决方案，但是它们的社区规模要比 PHP 社区小的多。当然，Web 服务器行业有容纳多个产品的空间，我不认为 PHP 会被所有的网站所使用。

记者：是否可以说社区的开源产品正在影响 Zend Server 免费版中的功能？举例来说，我认为 PHP6 将包含一个可选代码缓存标准，你们是否会对此推出相应版本？

苏拉斯基：这是我们决定增加的功能之一，但是它不是唯一的功能。

可选 PHP 缓存(APC)将成为 PHP 的标准，但是它不会带来多大变化，它已经在 PHP 扩展库中，用户可以非常轻松的安装它，而且如果用户关注 PHP6 的话，将会发现它并不会默认被启用。如果人们真的喜欢 APC，可以禁用 Zend Optimizer 插件然后使用 APC，除了小部分用户界面是 Zend 专用之外，其它完全相同。

记者：PHP6 的开发时间似乎过于长了。考虑到 Perl6 的情况，难道数字 6 是一个魔咒？或者只是计划的一部分？

苏拉斯基：或许吧，不过我认为 PHP6 将在 Python6 之前推出。PHP6 是一个比 PHP4 和 PHP5 都更难的项目，原因有二：一是 PHP 的代码库现在已经如此巨大；二是每一处细小的兼容性故障都会成为很多人的噩梦。而且由于 PHP6 将引入对 native Unicode 的支持，不可避免的会给该语言带来大量兼容性故障。坦白的说，我不知道最终结果会如何。

记者：与从 PHP4 转向 PHP5 相比，从 PHP5 转向 PHP6 的难度如何？

苏拉斯基：从版本 4 到版本 5 的迁移是相当成功的，只花了几年的时间，但是今天的 PHP5 已经比当初的 PHP4 更加普及。我们决定不着急进行升级，因此我们现在更关注 PHP5.3。

我们决定在 PHP5.3 中增加最初计划在 PHP6 中推出的一些新功能，诸如命名空间，这样我们不必急于升级到 PHP6。PHP6 正式推出可能还需要一段时间。

Mozilla 志愿者开发模式被复制

Mozilla 主要依赖开发者志愿模式来开发火狐浏览器，现在许多营利性企业也开始这样做，比如谷歌和 LinkedIn 都在努力复制 Mozilla 模式。

在 Mozilla 于 6 月 30 日发布的火狐浏览器之中，有一个很酷的视频播放器。但是对于某些视力不好的用户来说，播放器屏幕上的控制按钮稍微偏小了一点。因此，41 岁的 Mozilla 志愿者 Ken Saunders 自己开发出一款工具，让视力不佳的用户可以更容易使用那款播放器。对于 Mozilla 而言，Saunders 只是无数志愿开发员中的一员。虽然 Mozilla 的内部员工已经由 2005 年的 15 人增加到了 250 人，但是仍有 40% 的工作是由志愿开发者完成的，那些工作的内容包括编写程序代码、设计火狐图标等等。



现在，越来越多的企业开始学习 Mozilla 的模式，尝试通过那种模式去获得软件开发商和产品、品牌或某个概念的爱好者的集体智慧。火狐开发员 Mike Beltzner 指出：“这里面有一个结构，但是同时你允许人们去创新和探索，让他们自由地做他们想做的事，那就是更容易滋生创新的方式。”

在火狐的开发过程中，Beltzner 将它称作“从后领导”。他的团队只负责制定最高的方向性决策，比如每一个新版本火狐的发布日期等等。而 Mozilla 的员工和志愿开发商则逐个明确需要完成的具体任务，并赶在最后期限之前完成那些任务。在那些志愿者之中，已经形成了一个识别系统。那些志愿者被称为模块拥有者，拥有管理某些领域如布局的权力。

许多公司都想模仿 Mozilla、在线百科全书维基百科以及开源操作系统 Linux 的开发模式。《人人参与：无组织的组织力量（Here Comes Everybody: The Power of Organizing Without Organizations）》一书的作者 Clay Shirky 表示：“复制 Mozilla 的模式并不容易，但是我认为将会有越来越多的公司想方设法地激励用户参与进去。”

并非所有的努力都很顺利。商用网络网站 LinkedIn 在 6 月对 1.2 万名自称翻译的用户进行了调查，想弄清楚到底是什么因素激励他们帮助将网站内容翻译成其他语言。有 18% 的用户表示，他们那么做只是出于好玩的心理，大约有一半的用户希望他们的工作得到某种形式的认可，也有许多用户对于这项调查表示不满。美国翻译者协会（American Translators Assn.）甚至致信 LinkedIn 的 CEO，声称那项调查的目的被误导了，而且会引起许多麻烦。LinkedIn 公司发言人 Kay Luo 表示，公司并不是想要求翻译者免费劳动。她说：“我们调查的目的是想弄清楚他们的兴趣有多大。”

谷歌最近公开呼吁艺术家允许它将他们的作品用于网络浏览器的皮肤之中，结果受到了类似的指责。谷歌本意是不想付费，但是它却强调说这样做可为艺术家们提供展示作品的机会。谷歌在声明中说：“我们相信，这些计划可以为艺术家们提供一个令人振奋的机会，一个可以让他们将自己的作品展示在数百万人的眼前的独特机会。”

让人们免费贡献出自己的劳动，这对于打着非赢利性基金会的名号的 Mozilla 来说要相对容易一些。此外，Mozilla 模式还有许多可供其他公司学习借鉴的地方。IDG 的网络开发经理 Kevin Gerich 表示：“如果一家公司能够保证造福社区以及开放性，那么即使它是赢利性质的公司也没什么关系。”

火狐 3.5 最大的技术突破来自外部开发者。Mozilla 曾经希望火狐浏览器中增加一个让用户上网但不留下历史浏览记录的新功能，但是由于开发者们没能开发出那样的功能，Mozilla 就放弃了那个想法。但是就在火狐 3.5 发布的前夕，一位志愿开发者拿出了关于这项功能的一项计划，Beltzer 认为那个计划非常完美。因此新版火狐中才增加了保密浏览模式。

引领网络浏览器市场的微软 IE 浏览器也有保密模式，但是它并不包括火狐中的某些功能，比如从某个具体站点开始逆向清除浏览历史记录的功能。分析师们认为，正是这种精细巧妙的设计让 Mozilla 火狐在浏览器市场的竞争中脱颖而出。Forrester 的分析师 Sheri McLeish 表示：“现在，Mozilla 绝对拥有一定的竞争力。它们并非营利性组织，但是它们的产品所提供的理念却引起了许多人的共鸣。”据数据跟踪研究机构 Net Applications 称，今年 5 月，火狐在浏览器市场的份额由一年前同期的 19% 增加到了 22.5%。同期，IE 浏览器的市场份额下降了 7 个百分点，降到了 65.5%。

致力于技术协作研究已有数年时间的 Shirky 表示，人们之所以愿意为 Mozilla 那样的组织做贡献主要有 3 个原因：符合他们的兴趣、他们可以得到认可以及他们可以遇到兴趣相投的其他人。虽然几乎所有人都能找到具有前两个特点的活动，但是互联网在提供第三点上发挥着非常重要的作用。Shirky 说：“如果你向其他人说起协作社区，你会不断听到别人这么说‘我发现其他人感兴趣的東西正是我所感兴趣的東西。’”

Mozilla 志愿者 Saunders 就是最好的例子。在过去的 4 年里，他在火狐某些功能的开发上作出了巨大贡献，他每天都会与生活在澳大利亚、爱沙尼亚和密执安的志趣相投的其他志愿者联系。他说：“可以说，我们是同事。”

开源协议 GPL 份额下降 GPLv3 增长

随着开源软件在企业中的使用率的提高，更多的开源项目正在倾向更新的 GNU GPL 开源协议。

部分开源项目基于 GPLv3，黑鸭子软件统计大约有 9500 个基于 GPLv3 的应用，去年是 2345 个，相比去年数量已经增长了 4 倍，份额达到了 5.10%。在黑鸭子列举的使用的开源协议 TOP10 中，GPLv3 排行第四，仅落后占据 6.32% 份额的 BSD2.0。Artistic 协议份额为 8.68%。

GNU GPL 下降

同时 GPL 的份额由去年的 70%，下降到今年的 65%。GPLv2 份额是 50.06%，毫无疑问，即使去年的份额降低了 5 个百分点，仍不可忽视。Linux 内核仍然是 GPLv2，一段时期内，该协议不可能失去主要地位。

看起来更多的企业预算转向了开源，令人关注的是开源协议影响着开源软件如何使用的问题。很有趣的事情，GPL 的开源协议份额在走下坡路，而微软的开源协议竟然是社区冉冉升起的新星。

诺基亚欲使用 Qt 技术构造 Maemo/塞班一体化开发平

诺基亚称下一版本的 Maemo 移动版 Linux 操作系统（代号 Harmattan）将使用 Qt 而不是 GTK+ 作为应用程序的开发框架。按诺基亚平台产品开发部门的经理 Quim Gil 的说法，使用基于 Qt 的框架将便于开发者编写同时适用于 Maemo（诺基亚平板电脑操作系统）和塞班（诺基亚智能手机操作系统）的应用程序。Harmattan 是即将面世的 Maemo5（代号 Fremantle）的下一代操作系统产品。

按 gil 的说法，诺基亚不会在 Fremantle 中使用 Qt 技术，不过在开源社区这项技术将得到支持，Qt 技术是诺基亚去年收购了 Trolltech 公司后得到的。目前诺基亚的 Maemo 操作系统基本都是基于 Gnom 的，而 Gnom 使用的则是 GTK+ 技术。

目前，诺基亚正在邀请 Gnom 社区的开发人员与 Maemo 社区的开发人员一起协作，以便让基于 Gnom 的程序可以在 Qt 环境下正常运行。



Gil 还称诺基亚正在积极准备将 Maemo 进一步推向主流市场。不过，目前使用 Maemo 的设备如诺基亚 N810 平板电脑等，面向的主要是高端小型机市场，而这类普及度较低的产品对开发者则普遍缺乏吸引力。

为此 Gil 提议：如果我们可以用 Qt 设计出一套通用 API，并使这套 API 可以在 Maemo 和塞班上通用，那么开发者们就不会顾此失彼了。如果他们觉得 Maemo 用户群不够，那么可以把这款游戏很轻松地移植到塞班平台上，这样用户群就会大为增长。

他同时还表示，虽然诺基亚正在雄心勃勃地开展 Qt 计划，但他们同时也并没有放弃 Gnom 项目，并会一如既往地支持 GTK+。

另外，本周一诺基亚否认了先前外界猜测他们会启用 Android 操作系统的传言，如果诺基亚真的开始使用 Android，那么他们将需要同时面对 Android，Maemo 以及塞班三套开源操作系统。

印度政府欲自主研发处理器

据印度《经济时报》今日报道，印度政府将召集顶尖工程师共同设计一款名为“印度处理器 (India microprocessor)”的产品。

该项目的目的之一就是避免军事、通信和航空系统采用商用处理器而产生的安全隐患。这款处理器的设计将由一家名为 Zerone 的创业公司负责，该公司最初的 2 亿美元投资来自印度政府。该处理器将由来自多家印度科研机构的工程师共同设计，管理工作则由印度信息技术部负责。

印度《经济时报》援引匿名消息人士的话说，该项目的决议草案将很快呈交印度内阁大臣，以寻求项目资金。该草案提到：“除非印度拥有自己的微处理器，否则我们永远无法确保网络和武器不受威胁。”

据悉，设计师有可能会采用 Sun 的 OpenSparc 处理器设计技术搭配 Linux 操作系统和 MySQL 开源数据库软件。OpenSparc 是 Sun 的 UltraSPARC T1 和 T2 处理器的开源版本。另外，印度政府还对今后能否继续使用国外芯片技术表示担忧。这也是促使其创建 Zerone 的原因之一。目前 Zerone 已经招募了 400 名芯片设计师。

印度半导体协会主席普尔尼玛·舍诺伊(Poornima Shenoy)表示，自主设计芯片是为了解除国家安全问题的担忧。他说：“此举还将推动与芯片设计相关的本地化产业的发展。”

一名熟悉印度政府项目的芯片产业高管则认为，印度政府的这一计划存在瑕疵，他说：“这只是重新实施以前那些无果而终的计划而已。过去，当人们向印度政府提供英特尔 x86 技术和 Sun 的 Sparc 技术时，他们甚至都懒得回复。”

这名高管还表示：“我们为何不能使用 x86 技术来开发？或者利用 ARM 的核心来开发芯片？要设计一个芯片根本不需要 2 亿美元。从 Wipro 这样的本地公司找几个工程师就能设计出来。一个政府部门如何保持一项处理器技术的活力并为它提供支持？跟以前一样，该项目根本不会有任何结果。”

印度的芯片设计行业主要为海外客户从事外包设计。因此，这些企业在设计过程中不享有任何知识产权。

之前，印度也曾尝试过一些芯片项目，但均以失败告终。例如，“晶圆城(Fab City)”项目原本计划将印度南部城市海得拉巴打造成为自主晶圆工厂聚集的工业区，但多数工厂最终转型为太阳能项目。

据报道，Zerone 将通过出售微处理器并通过提供支持获得收入，同时还将为芯片设计师提供培训。印度境内没有大型芯片制造工厂，但该报道并未指明，印度政府是否会考虑通过海外工厂来生产这款处理器。

微软向开源示好：发布 2 万行 Linux 设备驱动代码

7 月 21 日凌晨消息，微软公司宣布首次直接面向 Linux 社区发布 2 万行 Linux 设备驱动程序代码，以增加其在虚拟化市场的竞争力。

首次发布 2 万行 Linux 驱动代码

北京时间 7 月 21 日凌晨，微软公司对外宣布，面向 Linux 内核社区发布 2 万行的设备驱动程序代码。这是微软首次直接面向 Linux 社区发布 Linux 设备驱动程序代码。这些代码的许可证类型是

GPLV2(GNU 通用公共授权第二版), 这是目前 Linux 社区最受欢迎的许可方式。

微软发布的 2 万行设备驱动程序代码可供 Linux 社区和客户使用, 这使得任何版本的 Linux 都可以基于微软的虚拟化技术 Hyper-V Server 2008 或者 Windows Server 2008 R2 Hyper-V 上进行虚拟化。在此之前, 因为双方有合作, 微软只提供了 Novell 公司的 Linux 操作系统使用其虚拟化技术的解决方案。

微软公司平台战略高级总监 Sam Ramji 在一份公开声明中表示: “向 Linux 内核社区发布 Linux 设备驱动程序是一次重要的里程碑事件。”

微软借此增强虚拟化市场

微软此次发布的 2 万行 Linux 设备驱动程序代码, 首先受益的是那些计划使用虚拟化技术的公司。在此之前, 用户如果想在既有 Linux 又有 Windows 的异构环境下使用虚拟化技术, 微软提供的解决方案只针对一家 Linux 厂商 Novell, 现在, 所有的 Linux 版本均可使用微软的 Hyper-V 来虚拟化 Windows 及各种版本的 Linux。客户有了更多的选择。

微软发布 2 万行 Linux 驱动程序代码也是其互操作性战略的延续。微软在三年前成立了开源技术中心(OSTC), 通过发布 Linux 驱动程序代码, 增强了 Windows 与 Linux 之间的互操作性。

对于微软而言, 它向竞争对手 Linux 提供帮助并不是一个亏欠的买卖。微软意识到, 虚拟化的趋势不可阻挡, 微软也对 Hyper-V 带来的收入寄予厚望。Linux 驱动程序的发布, 让所有版本的 Linux 都可以使用 Hyper-V, 业内分析认为这一举措可以提高微软在虚拟化市场的竞争力, 提升 Hyper-V 带来的收入。

Intel 成第二大 Linux 内核代码贡献者

刚结束的 Linux 专题大会上, LWN.net 创办人 Jonathan Corbet 公布了有关 Linux 内核代码开发者的统计数据 Red Hat 连续三年成为内核代码主要开发厂商, 而 Intel 则首次成为第二大代内核代码开发厂商, 主要是 Intel 正大力推动 Moblin 操作系统, 同时亦代表着 Intel 看好未来 Linux 操作系统前景。



一直以来, Red hat、IBM 及 Novell 均为 Linux 内核代码的主要开发者, 2008 年排名分别为 Red hat 11.2%、Novell 8.9% 及 IBM 6.1%, 这三家厂商主要针对商用市场, 但 2009 年出现了极大的改变, 由于 Intel 希望力拱 MID、UMPC 及 Smartphone 市场, 这个市场不一定需要采用 Windows 操作系统, 因此 Intel 大力推动基于 Linux 架构的 Moblin 操作系统, 因此在 2009 年大量

地提供了全新 Linux 内核代码。

因此，2009 年 Linux 内核代码的主要开发者排名，虽然 Redhat 仍然处于首位占 12.3%，但 Intel 已提供至第二位占 6.9%，IBM 及 Novell 则占 6.3%及 5.9%，分别排名第三及第四位。

据 Linux 首席技术长 Dirk Hohndel 表示，Intel 大幅提升对 Linux 的开发，标志着 Linux 操作系统的重要性不断提升，已不再局限于商用及服务器端的应用，尤其是 Intel 与 Google 投入了大量的资源，开发 Moblin、Android 及 Chrome 操作系统下，预期未来 Linux 操作系统在消费端 PC 的市占不断提升。

红帽、Novell 和 IBM 这些厂商都有大量的软件业务，因此对 Linux 进行开发不足为奇，而身为硬件厂商的 Intel 如此大规模开发开源 Linux 就有些奇怪了，根据 LWN.net 创始人 Jonathan Corbet 在渥太华 Linux 专题报告会上的最新统计数据，Intel 在 2009 年贡献了 6.9%的 Linux 内核，2007 年和 2008 年分别为 2.3%、4.1%。

尽管红帽仍然是当仁不让的冠军，然而按照 Intel 这两年的发展趋势，明年红帽就不一定稳坐冠军了，为什么 Intel 要如此努力开发开源软件？其 Linux 和开源首席技术官 Dirk Hohndel 在接受采访时解释说：“这是 Linux 社区增强的一个标志，现在很多方面都对 Linux 社区贡献自己的力量，这表明 Linux 是多么地重要。”不过，这个解释太简单也太牵强了。

Linux 最新 0day 的幕后故事和技术分析

我们应该知道不存在绝对完美的安全，没有一个绝对安全的程序。总之，安全是一个过程，而不是一个产品。Linux 在过程方面的安全是相当出色的，这也是它极大安全的原因。这有一个例子。



7月16号，一个名叫 Brad Spengler 的安全方面的程序员，他设计了一个开源网络和一个叫 grsecurity 的服务器安全程序，这个程序全面披露了安全邮件列表——Linux 内核 2.6.30 版的安全漏洞。

根据 SANS 英特网风暴中心的说法，关于这个漏洞比较简洁的表述是：“在进行网上或大规模运算时，代码的脆弱就表现出来了。简言之，开发者初始化了一个可以为 NULL 的变量。并紧接着正确地检查这个新变量在附近几行代码中的值，如果是 0（NULL），他就返回一个错误。”

但是，从技术的观点来说这是有趣的地方。程序员这样写的代码看起来是没错的。只有在使用

gcc “编译器开始处理这个变量，优化代码，编译器会发现变量已经被赋值(如果是 0)并且从编译好的代码中移除。换个说法，编译器会把变量引入到二进制码，不存在源代码中。这将会引起内核尝试去读/写 0x00000000 的数据，这使攻击者能映射到用户区，最终破解程序。”

听起来很吓人，是不是？你看，因为它在底层工作，这个漏洞可以绕过 SELinux（安全增强的 Linux），突破其它 Linux 安全程序。但实际上，这完全没有那么的值得惊慌失措。首先，为了攻击一台这样的计算机，你必须先拥有一个很高的 Linux 权限才能使用开发工具。其次，你需要重新开始编写一个程序，并把利用代码加入其中。这不像在 Windows 下，你只要错误的点下某些网页或是打开一封带附件的邮件，你的电脑马上就会感染上恶意软件或是和僵尸软件等威胁。你几乎真要尝试用这个方法打开你的 Linux 计算机来消灭它。但是，这个漏洞也许怎么都不能发挥作用。你看，同样是相似的溢出问题，也许潜伏着更大的危险的——使用 PulseAudio 时（一个流行的 Linux 和 Windows 媒体服务器）发生了溢出问题——在六月被修复了——这是重点。所以，如果你使用最新版的 Linux 内核，Linux 2.6.31-rc3，你应该感觉极好。

无论如何，事情的关键点并不在于技术细节。而是在于对开源软件，问题很快被大家发现并且被修复。举个例子，周三时，Firefox 3.5 有重要的安全漏洞被揭发，但周五时，Firefox 3.5.1 就修复了这个漏洞。

在周一，微软的 IE 浏览器同样有一个主要的安全漏洞被披露，但现在还没有看到补丁出现。这不足为奇。Windows 和其他微软程序的有些安全漏洞在几个月、几年后仍然没有补丁发布。至少，在去年年末，微软修复了一个存在了七年的漏洞——我不骗你。而且，让我们不要忘记，一个叫 MyDoom 的恶意软件——最早出现在 2004 年，到现在还可以在 Windows 电脑上用来进行 DDoS（分布式拒绝服务）攻击。

简而言之，所有开源软件比同类的商业软件更安全，因为开源的特性使安全漏洞的查找和修复快速得多。如果微软真关注它的软件安全，那么它在周四就应该对漏洞进行修补，它应该开放代码，从而是安全问题得到控制。可是，这种事情是不可能发生的。所以，现在和将来，FOSS（自由及开源软件）都会在安全方面更胜一筹。

云操作系统让思科与 Red Hat 亦敌亦友

什么是云的操作系统？Linux 厂商 Red Hat 可能会说是 Linux。但是网络巨头思科可能会给出一个完全不同的答案——尽管思科在很多领域都与 Red Hat 有紧密合作。

Red Hat 和思科上周都在忙于他们的自己的云项目。Red Hat 公布了一项新的云认证计划，而思科则透露了他们进入市场的云策略。那么，这两家厂商和他么的云项目之间有什么联系吗？

这个问题的答案强调了开源技术和 Linux 操作系统在云正在成为操作系统厂商下一个战场过程中所扮演的角色。

Red Hat 上周公布了针对其付费的弹性云服务的认证计划。到目前为止，Red Hat 已经将 Amazon 作为他们的一个认证合作伙伴。但是思科，尽管提供了一系列基于云的解决方案，但是目前并不是 Red Hat 认证的云服务提供商合作伙伴。思科方面对此作出的解释是，他们对与 Amazon 这样的提供商竞争并不感兴趣。

而且思科也不想将自己锁定在一个操作系统环境内。对思科来说，选择用于云环境的操作系统并不一定要是 Pro-Linux 或者 Pro-Windows。思科首席技术官 Padmasree Warrior 表示，思科的云策略就是以标准为驱动，希望用户有多重选择。Warrior 表示："我们不希望这种网络功能性只支持一种云操作系统。从这一点来说，我们是中立的。"



Warrior 还表示，思科将确保他们的云技术能够与多种操作系统兼容，包括 Red Hat 和它的 Linux 产品。Red Hat 仍将在其他支持云技术的领域保持与思科的紧密关系。

Red Hat 企业开发副总裁 Mike Evans 表示："我们与思科在多个领域有着稳定的合作关系--他们的 IT 部门是 Red Hat 一个主要用户，他们的很多设备和网络都嵌有 Red Hat 技术。而且，我们也为他们最新的 UCS 系统提供了主要的操作平台。"

UCS 系统将应用服务器集成到思科的网络设备中。Red Hat 是思科 UCS 系统的主要合作伙伴之一，目前至少有 20% 的早期 beta 测试用户都将 Red Hat Enterprise Linux 作为他们的 UCS 应用操作系统。

思科的云操作系统？

除了确保用户选择之外，Red Hat 还是思科 WebEx 交付基础架构的主要合作伙伴之一。思科软件合作群组高级副总裁 Doug Dennerline 表示："我们在 WebEx 方面与 Red Hat 有很多合作。"

Dennerline 还表示，Red Hat 是思科即将在今年年底推出的电子邮件产品--WebEx Mail 的"核心"。这项最新的云托管电子邮件服务是基于思科收购 Linux 电子邮件解决方案提供商 PostPath 所获得的技术。思科涉足开源技术领域也有一段时间了，而且思科自己就是开源技术的贡献者和消费者。Dennerline 表示，思科和开源技术之间的关系将在 WebEx 策略中继续下去。

他说："WebEx 以一种很有意义的方式将我们带入到开源世界。实际上我们现在调动了思科的所有软件资源，随着我们进入到这个市场中，我们将变得越来越开源，Red Hat 就是我们在这方面的合作伙伴之一。"

社区扫描

Linux 基金会推 Visa 白金信用卡促开源发展

北京时间 7 月 28 日晚间消息，据国外媒体报道，Linux 基金会周二推出了一款 Visa 白金信用卡，使 Linux 爱好者可以以另一种方式支持 Linux 操作系统。



Linux 基金会致力于推动 Linux 内核的发展。该基金会执行主管吉姆-泽姆林(Jim Zemlin)在一份声明中表示，用户可以以多种不同的方式为 Linux 做贡献，包括撰写代码和市场推广等。新款信用卡卡面上绘有 Linux 吉祥物；企鹅 Tux，将成为 Linux 支持者的“身份证”。该信用卡还提供一些优惠措施。

这款信用卡是 Linux 基金会和 CardPartner 合作通过 UMB 银行推出的。对于每张被激活的信用卡，Linux 基金会将获得 50 美元收入。该基金会还将从用户的每笔消费中获得 1% 的提成。

该基金会表示，从信用卡项目中获得的 100% 收入都将被用于举办社区技术活动，并为开源社区成员提供差旅补助，以加速 Linux 的创新。

这款信用卡最初将只面向美国居民推出，预计 Linux 基金会未来几个月将拓宽申请者范围。

不堪 Linus 批评 Alan Cox 选择退出

英国程序员 Alan Cox 是一位资深的 kernel 黑客，曾被视为是 Linus Torvalds 之后的二号人物，2009 年 1 月他从工作了 10 年的 RedHat 公司离开，加入了 Intel 公司。

除了维护 Linux kernel 子系统 TTY 之外，Alan Cox 还参与了 GNOME 和 X.Org 项目。2009 年 7 月 28 日，在受到了 Linus Torvalds 的严厉批评之后，Alan Cox 宣布退出，称“我已经受够了。如果你认为问题很容易修正，你去修正好了。我已经压缩了 tty 合并队列，任何为 tty 层提供补丁的人可以将它们发送给新的维护者”。

管理员最终出现 CentOS 项目危机化解

CentOS 是一个建立在红帽企业 Linux 基础上的免费提供源代码的 Linux 发行部，目前已经触及了重大冲击的道路：该项目的主要管理者，兰斯戴维斯目前失踪中。这是怎样的一个问题，因为戴维斯是 CentOS.org 域名、IRC 频道和 CentOS 资金唯一的管理员。



其他 CentOS 小组成员在一封公开信中详细介绍了项目的困境：

这是 CentOS 开发组成员给给兰斯戴维斯的一封公开信，令人遗憾的是，我们不得不发出这封信，但我们没有其他选择。一段时间以来，我们一直在试图解决这些问题：您似乎已经爬进了一个洞...这是不能接受的。

您长期承诺的 CentOS 计划资金没有如期到来，您是 centos.org 域名的唯一管理员却没有承担起应有的责任，作为 IRC 频道和 CentOS 资金管理员，一样如此。在过去的两个星期内，当我们尝试呼叫你在英国专用的电话号码时，总是提示线路繁忙。直到昨天，才有你摘机的语音邮件声音，和我离开的消息迫切要求答复。Karanbir 报告还呼吁，并保留邮件没有收到您的回复。

请不要让你所管理的 CentOS 项目就这么消亡。很显然，一旦所有开发者都离开，那么这个项目肯定就死了。请与我联系，或者回复此邮件，安排必要的工作，以保持该项目和 centos.org 域名持续进行。

但是事情很快又了转机，在美国当地时间 8 月 1 日，CentOS 项目管理组在其网站上发表声明，他们已经和兰斯戴维斯取得了联系，并且达成了某些协议，项目将不受影响，继续保持良好的发展势头。网站声明如下：

CentOS 开发团队已经在今天举行了一个例行会议，Lance Davis 也列席参加。在本次例行会议中，已经就上次公开信中提到的很多问题达成了共识，没有解决的问题也已经做好了计划。所以，CentOS 的任何用户 请不用担心，CentOS 项目将一如既往的存在并发展下去。

目前 CentOS 项目已经控制了 CnetOS.org 和 CentOS.info 这两个域名，以及这 CentOS 项目下的所有和项目相关的商标、资料和设计资料等。我们将很快和 Lance 就相关没有解决的问题尽快达成协议。

更多信息，我们将尽快提供。

SourceForge2009 社区选择奖名单公布

SourceForge 组织的 2009 社区选择奖评选完毕，今日在其官方网站公布了获奖的开源软件项目。名单（奖项、项目、投票率）如下：

最佳系统管理工具：Mysql 管理工具 PhpMyAdmin 18.82%；

最佳学术项目：思维导图软件 Xmind 21.54%；

最可能改变做事方式的项目：PortableApps.com: Portable Software/USB 20.21%；

最佳多媒体项目：多轨音频编辑器 Audacity 32.09%；

最佳开发者工具：代码编辑器 Notepad++ 19.21%；

最佳视觉设计：Best Visual Design 27.87%；

最佳游戏项目：游戏模拟器 ScummVM 20.38%；

最佳企业项目：数据库服务器 Firebird 18.05%；

最佳政府项目：办公套件 OpenOffice.org 48.27%；

最佳商业开源项目：PortableApps.com: Portable Software/USB 41.74%；

最佳新项目：迷你电脑操作系统 eeubuntu 27.52%；

最佳项目：PortableApps.com: Portable Software/USB 16.45%；

其中 PortableApps.com: Portable Software/USB 无疑是大赢家，获得了最佳项目、最可能改变做事方式的项目、最佳商业开源项目三项大奖。

用户通过 PortableApps.com: Portable Software/USB 可以将自己喜爱的应用程序装到 USB 闪存盘或外部硬盘。安装后，应用程序就可以在任何 Windows 计算机上运行。任何个人的数据随身携带，十分方便。

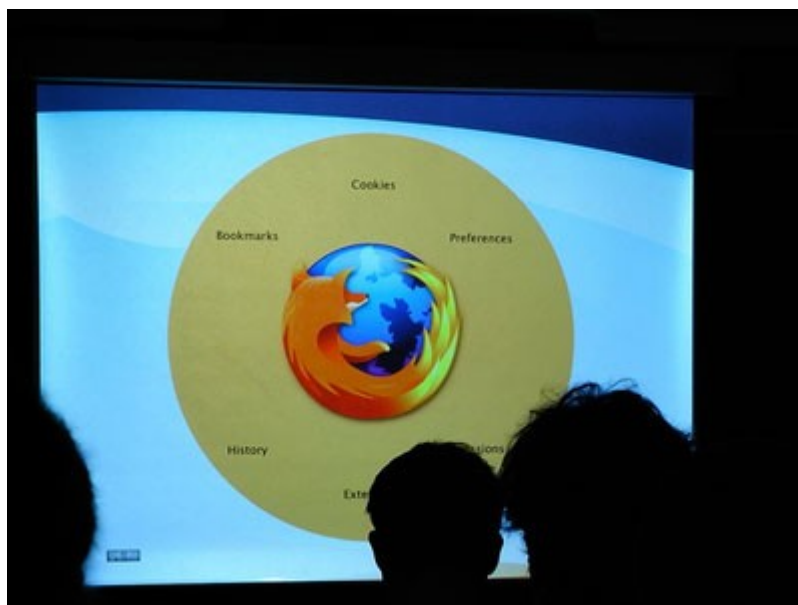
Debian 采用两年的发行周期

多数流行的发行版如 Ubuntu、Fedora 和 Mandriva 都采用固定时间的发行周期，而以发行周期漫长而著称的 Debian 也开始随大流，它正式宣布了发行日程表。与其它发行版通常采用的 6 个月时间不同，以稳定性闻名的 Debian 采用了两年的发行周期，单数年的 12 月份开发冻结，后一年的上半年发布正式版。

根据新的计划表，Debian 6.0（代号 Squeeze）将于今年的 12 月冻结技术，明年上半年发布。考虑到 Debian Lenny（5.0）是在 2009 年的情人节发布的，新计划表显然让 Squeeze 的开发有点紧迫，它将是两年周期的一次例外。对于需要长周期升级的大型机构和用户，Debian 项目也提供了跳过升级的可能性方案，允许他们直接从 Debian GNU/Linux 5.0 升级到 Debian GNU/Linux 7.0（尚未有开发代号）。虽然开发时间短暂，Debian 仍然希望在 Squeeze 中实现几大目标，包括多架构支持，让系统的安装变得更轻松；以及优化启动进程，改进启动性能和可靠性。

Firefox 3.5 的启动堪比操作系统

一位 Windows Firefox 3.5 用户发现，仅使用了一天之后，第二天他的浏览器启动时间长达 35 秒。他试图找出浏览器花如此多时间启动的原因，他发现在 Mozilla 论坛上有很多人遭遇了同样的问题。



根据论坛上透露的信息，启动缓慢与硬盘上有太多的垃圾有关，而罪魁祸首是 NSS(网络安全系统)，它需要执行大量的加密和安全相关的操作，需要使用到随机数。真“随机数”是很难产生的，NSS 的开发者没有使用系统提供的随机选择功能，而是创造性的发明了一种新的随机数生成方法——用多个线程读取硬盘上临时文件夹，把这些文件作为 seed 创造出随机数。在启动时，Windows 版的 Firefox v3.5 需要读取 Internet Explorer Cache 和 Windows 临时文件夹，还扫描了所有的 font 文件，于是 Firefox 启动变成令人痛苦的过程。

Ubuntu 积极推动开源发展

据国外最新资讯报道，Ubuntu 的社区经理培根就开源和专有模型话题公开发表了一次演讲表示，专有软件开发商在发展的尺度和协作方面，应该积极向开源公司靠拢。

在接受安全机构 Coverity 公司采访时，培根进一步提出规范 Ubuntu 管理的措施。他表示，规范的关键在于对开源软件开发人员加强管理和保持代码的安全和使用方面。目前，大部分 Ubuntu 代码的开发和维护工作已经具体落实到个别开发商团队身上，最近成立的一个专门的安全开发团队就是典型的案例。

培根强调，就当前的形势来看，Ubuntu 正以势不可挡的速度向前发展，时刻确保操作系统的稳定和安全就要求我们必须寻求更多开发人员的协作和支持。

在演讲中，培根用“多眼球的代码”的哲学来形容开源的发展模式，赞美之词溢于言表。他表示，在私有开发环境里经营的开发商，如果能从开源软件中获得利益，他们就一定会大力倡导使用开源软件。

而从技术代码的质量角度来看，专有软件公司则可以从开源模型中学到很多东西。比如，目前，

一个社区就可以借助一个坚实的捐款手段和 onramp 的手段，去有效地改进代码的质量。

最后，培根明确表示，当今世界上，越来越多的开源项目、平台和产品相继出现，开源的用户可以从中挖掘出很多的商业服务机会，因此，开源公司成长的远景是扣人心弦的。

巴西总统头戴 ODF 帽现身自由软件会议

巴西总统卢拉最近出席了在巴西阿雷格里港举行的 FISL 10 会议，他发表讲话强调了开源软件对巴西的重要。会上他戴上了印有 ODF 标记的帽子。

完成大会的公开发言后，卢拉还向一个较小的团体发表了讲话，其中包括自由软件运动创始人 Richard Stallman、自由软件社团领袖 Jon Maddog Hall、巴西自由软件社团领袖 Pablo spectra Lorenzoni、巴西前 IT 主管 Amadeu da Silveira、惠普公司开源事务主任 Bdale Garbee 以及 Red Hat 副总裁 Michael Tiemann。坐在前排的 Jon Maddog Hall 称，卢拉的讲话持续了大约 15 分钟，并且完全脱稿，显示了他对开源软件这个话题的熟悉。

Oracle 可能将终止 OpenSolaris 项目

当甲骨文完成收购 Sun 的程序之后，它会选择抛弃哪一个项目，很多人推测是 OpenSolaris。



IT 业外人士很少会认为甲骨文公司是一家 Linux 企业，但它确实是。甲骨文不仅仅努力向消费者推广它的 RHEL (Red Hat Enterprise Linux) 克隆版 Oracle Unbreakable Linux，并且在企业内部甲骨文也长期在服务器和部分桌面上使用 Linux。那么像甲骨文这样的 Linux 企业收购了 Sun 的开源操作系统后，会拿来干什么？答案显然是“Nothing”。内部人士透露 OpenSolaris 开发者非常担心，当 Sun/Oracle 合并完成之后，他们要么必须在 Linux 上工作，要么被迫离开。

Ubuntu：下一版本不一定采用 GNOME3.0

7 月 14 日 Ubuntu 首个 LTS（长期支持）版本 Ubuntu 6.06 结束了使命，LTS 版本每两年发布一次，桌面版本支持期限为 3 年，服务器版本为 5 年，在 Ubuntu 6.06 LTS 之后是在 2008 年 4 月发布的 Ubuntu 8.04，而下一个 LTS 版本有可能就是明年 4 月将发布的 Ubuntu 10.04。

在最近的 Gran Canaria 桌面峰会上，Canonical（Ubuntu 开发机构）创始人 Mark

Shuttleworth 就相关问题接受了采访，其中谈到了下一代 LTS 和界面 GNOME 3.0，下一代 LTS 可能是 10.4 也可能是 10.10 版本，这要看 Debian 和 Ubuntu 之间对“Meta-Release-Cycle”（目标发布周期）的讨论结果，这是 Debian 提出的代码冻结日期，代码冻结意味着主要的组件已经确定，其它基于 Debian 的发行版可以考虑协调发布。

至于下一代 LTS 版本中是否会采用 GNOME 3.0 界面很多人的意见不统一，有人说因为是 LTS 版本所以请在其中使用这个超酷超炫的 GNOME 3.0 吧，也有人表示 GNOME 3.0 架构只是一个可选的功能不要因为它而推迟了 LTS 版本的发布。

Shuttleworth 表示 GNOME 3.0 会如期完成，Ubuntu 不会刻意去改变什么，但是现在最重要的是讨论 LTS 版本中应该有什么，如果我们在 1000 万用户没有准备好的情况下就让他们使用一些功能那么对开发商非常不利，因此 GNOME 3.0 不一定会在下一代 LTS 版本中采用。此外 Shuttleworth 表示，虽然不敢保证下一代 Ubuntu 版本会采用新的主题，但是会在下一代 LTS 版本中提供全新的主题。

最新 Linux Kernel 发现 0day 安全漏洞

研究人员 Brad Spengler 发布了最新 Linux Kernel——2.6.30 和 2.6.30.1——安全漏洞的攻击代码，攻击者能利用 NULL 指针错误（NULL pointer dereferences）侵入系统，能绕过 SELinux、AppArmor 和 Linux Security Module 的保护。SELinux 是美国国家安全局开发的安全增强实现。

Spengler 表示这是他见过的第一个能被利用执行任意代码的 Linux Bug。他说，仅仅花了两小时，他便实现了代码执行，找到了如何关闭 SELinux、AppArmor 和 LSM 的方法。他公开了一段演示视频（Youtube），有缺陷的代码位于 net/tun 执行上。

开源社区协作工具 Launchpad 源码公布

开源软件发展的巨大的优势就是开源社区的参与者之间的交互，但是交互发生在项目之间如何解决？一个基于 Web 的社区协作工具 Launchpad 登场了。

Launchpad 公测版发布 2 年后的今天，Canoonical 公司公布了 Launchpad 的源码，将允许开发社区使用 Launchpad 工具。

Launchpad 不仅允许项目团队协作，而且方便在不同的项目间上传和下载等交互。该服务帮助项目通过一组 6 个工具集成组件来协作，包括团队管理、Bug 跟踪、代码托管、翻译、计划跟踪和答案跟踪。该功能的使用均可在 Launchpad 网站使用。

使用 Launchpad，开发者托管和分享代码免费，使用 Bazaar 版本控制系统集成到 Launchpad。另外，消除了协作的障碍，同时支持不同项目托管服务-Launchpad 实际上成为了一个社会化网络。

Launchpad 能够支持任何的软件项目开发，开源或者闭源。但是托管开源免费，托管闭源收费，也就是说 Launchpad 可以控制代码分享与否。

Oracle 为什么不会干掉 MySQL

曾经有人推断 Oracle 通过收购 SUN 进而取得近年表现得越来越勇的开源数据库 MySQL，从而可以顺理成章地将其处死，但 MySQL 的前任首席执行官 Mrten Mickos 却有不同的看法，相反 Oracle 急需 MySQL。

Oracle 首席执行官拉里埃利森可能会成为开源的朋友，因为 MySQL 可以用来对付 Oracle 的长期敌人微软。SUN 去年以 10 亿美元的价格收购了 MySQL，传言 Oracle 也是其中的投标人，不过现在 Oracle 却从 SUN 手中明显捡了个便宜(除了现金和债务外，Oracle 只给了 SUN 56 亿美元) 以下是上周一《财富》杂志对 Mickos 的采访记录。

财富：你对于 Oracle 收购 SUN 这一事件的第一反应是什么？

Mickos：有点吃惊，但不是非常吃惊，Oracle 公司有钱，也是一具有长期战略眼光的公司，它们进入应用程序领域与 SAP 展开竞争，从目前来看似乎已经成功了，现在它们又想与微软展开竞争了。

微软的数据库业务增长速度非常惊人，Oracle 可以使用 MySQL 拉拢规模非常庞大的开发者社区。但这并不会危害到 Oracle 的数据库业务，虽然 MySQL 的增长速度接近疯狂，但 MySQL 主要用于基于 Web 的应用程序，而 Oracle 主要用于老的，传统的应用程序。

不可否认 Oracle 可以干掉 MySQL，但我认为他们肯定不会这么做，拉里埃利森可是个聪明的人，当我离开的时候 MySQL 每天的下载量达到了 7 万次，对年轻开发人员特别有吸引力，对于 Oracle 来说 MySQL 业务发展有着重要的意义。

财富：这笔交易对 MySQL 开发者社区有何影响？

Mickos：任何流行的开发项目都有多个社区，他们不在乎谁拥有它，他们在乎的是社区是否能够使用，MySQL 的装机量已经达到了 1200 万，也许只有你和我会注意这件事情，但很多开发人员甚至不知道现在 MySQL 已经属于 SUN 了。

InnoDB(2005 年已将其收购)仍然非常流行，Oracle 可能会独立运作 MySQL，MySQL 的力量是它的开放性，而 Oracle 不同，它们没有公开它们的 bug 数据库，如果 MySQL 将来也不公开，那将会逐渐失去其市场优势。MySQL 的一个优势是它的商业模式，但我相信 Oracle 的管理人员能够懂得这一点。

财富：那拉里的反开源叫嚣是虚张声势吗？

Mickos：是的，我认为他已经喜欢上开源了，我只是猜测，但他是一个可以跳出盒子思考的思想家，他一定会用 MySQL 与微软展开有力的竞争，微软一直是拉里最喜欢的敌人。

行业观察

甲骨文需要面对的一个开源竞争对手：Ingres

MySQL 在 SUN 被甲骨文收购后，这一著名的开源数据库已经成为甲骨文庞大帝国的一部分，而曾作为甲骨文重要竞争对手的开源数据库，MySQL 在甲骨文全盘收购 SUN 之后，将慢慢的变成甲骨文发展计划中的一部分。曾几何时，Ingres 和 MySQL 是甲骨文在数据库软件市场的主要开源替代者。



换句话说，Ingres 正在成为开源世界中唯一抵抗甲骨文的最后堡垒。或许有人说，还有 PostgreSQL 和 Firebird，但是据市场分析机构 Gartner 称，Ingres 是企业关键系统领域的唯一开源数据库，PostgreSQL 背后并没有公司做支持服务，而 Firebird 则更适合于嵌入系统。

本周早些时候，Ingres 公司发表了一个声明称，它完全有希望成为甲骨文的一个可靠替代者。该公司透露，连锁超市 Save Mart 和纽约投资银行考恩集团已经选择了它的技术来取代甲骨文。Ingres 公司 CEO、纽约证券交易所前 CIO 罗杰布克哈特表示，“这表明人们对 Ingres 提供的替代方案具有足够信心，因此他们将关键任务系统迁移到我们的平台。”他表示，他希望看到 Ingres 采用率将迎来一个“滚雪球式”的增长，现在该公司已经公布了它取得的一些重大成功案例。

科恩集团 CIO 丹尼尔·弗兰克斯(Daniel Flax)选择了 Ingres 作为公司新程序交易门户的技术基础。他表示，购买 Ingres 技术的成本，约为购买传统数据库厂商技术的三分之一，而且，“从技术角度来看，它可以完成我们想要的功能。”

Ingres 仍然是甲骨文脚下的一个小鹅卵石。尽管它一直在快速增长，但去年的营收仅为 6800 万美元，简直无法和甲骨文公司的 270 亿美元相提并论。甚至昨天在甲骨文季度电话会议上，证券分析师都不屑于提到 Ingres。但是，只要布克哈特不断赢得新成功，至少企业会知道他们还有一个开源替代产品。

实际上，甲骨文收购 MySQL 给人们带来的焦虑或许会让 Ingres 从中受益。

Save Mart 连锁超市并没有将甲骨文列入其数据库选择的名单，尽管该公司目前正在使用着老版本的甲骨文数据库。

该公司 CIO 詹姆斯·西姆斯(James Sims)表示，“甲骨文对待我们非常糟糕，带有不屑和自大，我直接告诉他们：我们和谁合作也不和你们合作。”

于是西姆斯选择了开源数据库 Ingres，和其它商业化开源公司一样，Ingres 通过销售支持服务来盈利，尽管该公司规模相对较小，但其技术却足以完成 Save Mart 的系统。

西姆斯表示，公司的系统不大，但运行非常高效，他对在更大规模的解决方案中实施 Ingres 也充满信心。

科恩集团最初也曾考虑选择 MySQL 数据库，但由于考虑到 Sun 正在被甲骨文收购，MySQL 的未来存在很大不确定性，最终还是选择了 Ingres。

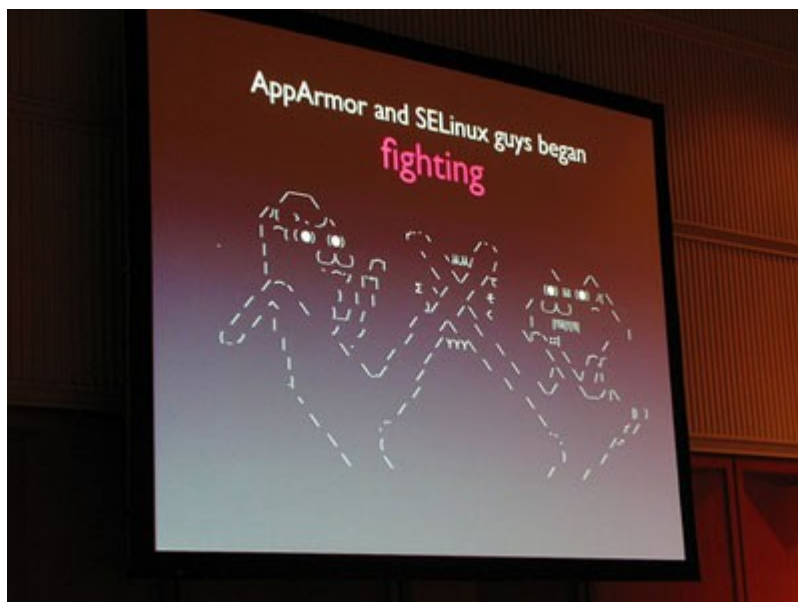
Monash Research 公司的分析师科特·莫纳什(Curt Monash)表示，开源数据库拥有着大量的

用户人群，而真正成规模的开源数据库并不太多，这样在“伙伴”MySQL 被甲骨文收购之后，Ingres 也迎来了向甲骨文直接叫板和发展的机会。

开源技术依然存在不足之处需要改进

你可以不爱开源技术，你可以不理开源技术，但是你不能不用到开源技术。在现今的 IT 社会，开源技术已经渗入了人们的生活，但是这么重要的开源技术依然存在不足之处需要改进。

“你们当中肯定有不少人对此感到有些吃惊，不相信这是真的，” Gartner 研究副总裁 Mark Driver 表示。“还有一些人会说，‘为什么要等到 2011 年呢？现在它已经来了。’”即使你不打算使用完全开源的应用软件，网络管理人员也应该对开源技术这个大趋势多加留意，因为某些商业软件供应商的开源选择可能会使用户暴露于风险之中或出于竞争劣势。“开源技术很快就将进入你的网络之中，不管你愿意不愿意，” Driver 说。“要想完全避免不受到它的影响是完全不切实际的。”



其实，开源技术并不像它的积极倡导者向你保证得那么好，也不向开源技术批评者所描述的那么危险，Driver 说。使用开源技术最重要的一件事就是制定一个开源策略，确定一个什么时候以及在什么地方使用开源技术的指导方针。很多企业的 IT 部门都纷纷制订自己的开源策略，但几乎没有一个全力以赴进行实施的，Driver 说。“如果你采用一种“既不问又不说”的策略监管开源技术的使用的话，我建议你还不如完全不使用它呢。” Driver 表示。“你知道它存在于你的企业里。如果你不能管理它，那么你就无法控制它，” Driver 说。

有些人认为如果使用开源技术的话，可以得到更好的质量保证并降低软件的总体拥有成本。“但是如果一个用户认为这始终是正确的，在将来某天肯定会陷入深深的失望之中，因为开源技术并非十全十美，” Driver 说。

一个企业是否采用开源技术的决定应该取决于四个因素：

第一个因素就是开源软件能否满足你的需要。这可能似乎是显而易见的，但一些开源技术的支持者却夸大了它的能力。“开源技术在企业中拥有大批支持者，一群非常狂热的支持者，” Driver

说。“这些人经常说‘Windows 不稳定并且不安全，我们应该使用 Linux。’”

第二个因素就是开源产品是否足够成熟，它的风险以及所带来的投资回报率是否可以接受。并且还要考虑开源软件是否有充足的服务和技术支持。

第三个因素是公司技术采用的态度。该公司愿意使用未经验证的新技术吗？该公司是否总是需要第三方支持？并且它能在开源软件支持上提供多少内部人员？

第四个因素是看看部署目的是不是用于关键任务。如果需要部署的应用程序需要时刻运行不能出现任何中断的话，那么你需要确保你要使用的开源软件符合你的需求。这并不是说开源软件不能用于关键任务。只是提醒你在将开源软件用于关键任务时最好小心一点，Driver 说。“现在，越来越多的保守的公司和关键任务解决方案都开始使用开源技术了，”他说。

客户可能会担心分裂问题，因为开源软件可以有很多不同的版本，比如 Linux。不过，Driver 表示，自然选择的形式能够防止开源社区的大规模分裂。“没有人能够保证 Linux 在将来会不会分裂，”某种程度的分裂是不会带来什么问题的，因为垄断的做法在开源社区并不可取。“自然选择能够清除竞争力比较弱的变体，也就是所谓的优胜劣汰。然而，专业化却允许变体并存，”Driver 说“因此，专业化是开源技术一个极其重要的特征。”

Driver 表示，他遇到的好多人都认为：免费提供的开源软件将会在未来 5 年内打败微软。Driver 认为这种想法真是太可笑了。开源代码正在改变商业软件这个行业，而这个行业本身也正在改变我们对于开源软件的看法。一些商业软件供应商采用技术来增加自身的竞争优势。“为什么 IBM 近来对于 Linux 持赞赏和支持态度？”Driver 问。“因为 Linux 的受欢迎度越来越高，直接影响到了 IBM 的主要竞争对手微软。”

在某些情况下，商业软件供应商正在破坏开源运行的理想。一些厂商使用一种 Driver 称之为“门源(gated source)”的模式推销自己的产品，也就是他们给客户适当的权力改变产品的代码，但只能供用户自己使用而不能被重新分配。在解释为什么要这么做的时候，基本上每个供应商都会说：“我会给你一些自由，但是这种自由绝对不能影响到我的商业利益，”这就是为什么要就清楚地界定什么才是真正的开源软件的重要性。Gartner 认为，一个开源软件只有被授予许可的时候，比如通用公共许可证(GPL)才能称得上是开源软件。

“我们不相信开源将会演变成另外一个营销词汇，”Driver 说。“这对于软件也将是一件坏事。”希望未来的开源技术还是我们熟悉的那个开源技术，而且会成为更加完美的开源技术。

谷歌 Android 进军企业市场之路漫长而曲折

美国第四大移动运营商 T-Mobile 星期二宣布，8 月份它将推出一款新 Android 智能手机，让人们感兴趣的是它能否具有更大的竞争力而被企业采用。新 Android 智能手机 myTouch 3G 将配置一个 3.5 英寸的触摸屏，350 万像素的摄像头和一个预装的 4GB microSD 内存卡。然而，这些面向消费者的功能或许还不足以满足企业在安全和任务整合方面的要求，这方面黑莓智能手机是个标杆。

IDC 高级分析师拉蒙·拉马斯(Ramon Llamas)表示，“现在市面上有很多智能手机，它们都具有创新功能。但是它们能否被企业接受是另外一回事。”

谷歌 Android 的优势

T-Mobile 的 myTouch3G 具有漂亮的外观。它的一系列新功能建立在广受用户欢迎的前任 G1 手机的基础上，不过新款 Android 手机使用一个虚拟键盘代替了 G1 的硬件键盘。新 Android 智能

手机让用户可以自定义菜单、壁纸和图标。

另外，成千上万的 Android Market 应用可让它与苹果 iPhone 应用商中的的大量应用相媲美；企业可能会发现值得选择 Android。此外，myTouch 3G 高度集成了谷歌服务，而且最新的 Android 软件可能让公司决策者更具备将其用于工作中的理由。

拉马斯表示，“新版 Android 与旧版 Android 类似，是完全面向个人用户设计的。在某些情况下，企业用户会有自己的设备——但是 Android 在这方面没有一个完整的解决方案。当然，这并不意味着企业不会选用 Android 手机，只是它依然处于初级阶段。”

Android 技术确实有被企业采用的潜力，但部署智能手机模型来取代或补充现有企业通信模型是一个缓慢的过程。



Forrester Research 负责新兴营销渠道、营销和广告、移动营销的分析师尼尔·斯特罗瑟(Neil Strother)表示，“由于更换硬件涉及到成本，决策者需要慢慢撤换设备。”他表示，从企业的角度看，更多的企业倾向于选择黑莓智能手机，而非苹果的 iPhone 或 Android 智能手机。因为黑莓手机可以为企业用户提供电子邮件功能，而且要比其它智能手机安全的多。

进军企业市场还需做什么

在部署手机到工作环境中时，IT 经历面临的一个最重要的考虑因素是网络安全。斯特罗瑟表示，未来黑莓手机竞争对手们存在的部分问题是，它们还没有被市场看做一个企业级解决方案。

他表示，企业 IT 人员与个人用户考虑的问题不同，安全是一个因素，是另外一个。在一个企业很难测试众多不同的设备。斯特罗瑟表示，在 Android 手机扎根于企业环境之前，企业职员可能已经开始普遍使用 iPhone。当需要达到一定程度时，IT 部门开始考虑这些新设备。

对企业来说，RIM 公司的黑莓手机仍然是无可争议的领袖。Frost & Sullivan 首席分析师格里·普尔蒂(Gerry Purdy)表示，截至目前，Android 主要是被个人用户使用。

普尔蒂表示，“Android 开发团队需要关注企业需求，关注安全。举例来说，它需要在设备上提供密码保护、加密和远程控制等功能。”

他补充说，无论是 iPhone 还是 Android 智能手机，要想更胜任企业任务，需要具备与 IBM 的 Lotus Notes 和微软 Exchange 同步的功能。这些功能是个人使用和企业使用的区别所在。

谷歌 Android 已开始努力

iPhone 和 Android 手机都已经开始向企业方向发展，但是这个过程可能是漫长且曲折的。

普尔蒂表示，在最近刚推出的 iPhone 3GS 中，苹果公司提高了安全性，并增加了一个主动同步(Active Sync)功能，使其更适合企业使用。

Android 智能手机开发者也已经意识到企业市场潜力巨大。举例来说，在 T-Mobile 的 myTouch 3G 中，谷歌让企业可以创建自己的应用程序。普尔蒂表示，“这对谷歌来说非常重要，说明 Android 已经开始解决这些问题。”

普尔蒂补充说，要想真正进入企业，“谷歌需要与 T-Mobile 这样的运营商和手机生产商合作，多方合作制定出一个切实可行的解决方案。”

外电评论：开源在中国面临重重困难

肆无忌惮的软件盗版和乏善可陈的商业支持妨碍了开源在中国发展壮大。

中国政府支持了多个国内的开源项目，但这些项目没有一个被广泛应用。分析人士称，鲜为人知，缺少大型的开源项目，难以找到某个程序语言的专门知识，所有这一切都妨碍了开源在中国的增长。

在中国，很少能听到有大型商业机构选择 Linux 或其它开源软件的事例。中国一直试图培育自己的软件行业巨人，包括开源领域的巨人。红旗 Linux 是中国最大的本地 Linux 供应商。红旗软件的 CEO 声称中国邮政和邮政储蓄银行的数万服务器运行的便是红旗 Linux，中国国有性质的媒体新华社也是该公司的客户。中国 Linux 市场的主要成长空间是上网本和服务器。但是 Linux 和其它开源软件在中国面临难以跨越的巨大障碍。

中国的企业并不需要选择开源软件以节省开支，因为他们有无数的盗版软件可供使用，盗版 Windows 和 Microsoft Word 基本上已经控制了中国的家庭和办公用电脑。商业软件联盟的报告称 2008 年中国的盗版率为 80%。北京 Linux 用户组组织者 Frederic Muller 表示，Windows 在中国相当于免费，任何闭源软件都是免费的。开源会计软件 FiveDash，虽然公司的代码开发是在中国完成的，但它却迟迟不愿意推出软件中文版。Sun 中国开源社区负责人 James Bai 表示，低知名度和 IT 行业缺少如 IBM 这样的大支持者，也是中国开源特色之一。

虽然中国也有除红旗软件之外的开源厂商，如 Linux 发行商中国标准软件公司（CS2C）和 RedFlag2000 Software，Office 组件开发商 Red Office。但是这些公司既没有促进开源社区的成长，也看不到创新。

谷歌的开源：窃取胜利果实，还是拥抱社群？

Google 表示将打造自己的操作系统，但全球各地热衷操作系统发展的人士可能只有一个反应：“拜托，别再生出另一种 Linux!” 这个说法是因为过去这 20 年来，Linux 与开源社群一再遭遇的问题，就是“分裂”。

从 1980 到 1990 年代，Unix 操作系统不断经历分裂，迫使系统管理员必须学习各种不同的平台：Solaris、AIX、HP-UX、FreeBSD...等等。

这已经够糟的了，不料脱胎自 Unix 的 Linux 问题更严重，四分五裂的程度犹有过之。视所在地点而定，这十年来你很可能已转换过多种 Linux 版本，看哪一种最盛行。

就个人而言，我一开始是用 Red Hat，但 Red Hat 后来分裂，分成正式的 Red Hat 版本和称为 Fedora 的开源社群版本。我短暂尝试了一下 Mandriva 以及 Suse，接下来几年改用 Slackware，然后又转向 Debian。这期间，我学到很多事，包括不同的套件管理、设定、开机和窗口管理系统。

当然，我也用过各种版本的微软操作系统，以及苹果 Mac OS X 和先前的版本。

经历过多年来 Linux 众多版本厮杀的混乱局面，现在我们然而，就在此时，Google 决定打造自己的 Linux 版本，将使 Linux 社群再度分裂。这么做只能以鲁莽和自恋称之。

Google 不自己另辟蹊径，而是打算利用 Ubuntu 创始人 Mark Shuttleworth 及其程序设计大军已经打下的江山。换言之，Google 打算用自己的马匹，来拉 Ubuntu 马车。



如果 Google 真的想“在 Linux 核心之上设计新的视窗系统(windowing system)”，没什么能阻止搜索巨人与这一行的顶尖好手公开合作。我相信，对于 Google 计划“彻底重新设计”Linux 基础安全架构一事，Linux Torvalds 会发表某种措词强烈的评论。Google 已承诺让 Chrome OS 开放源代码，一如 Google 对 Chrome 浏览器与 Android 手机操作系统所做的一般。

不过，这些计划仍引起一些疑虑。比方说，它们如何融入纯粹由社群支持与维护的开源计划？与 google 在线广告业务的关联性有多密切？

Android 是很棒的移动装置操作系统，仅次于苹果的 iPhone 平台。但大部分的 Android 开发仍由 Google 一手掌控。而且，Android 的使用者都同意，这个操作系统与 Google 的云端服务(例如 Gmail)密切整合。但若是你偏好使用 Windows Live 或其它操作系统，事情就可能变得棘手了。

Chrome 是很棒的浏览器，但也一样主要操控在 Google 之手。支持浏览器自由竞争的我们很庆幸，Mozilla Firefox 是完全独立的，不必为任何人的企图服务。

以后该信任谁？非营利性的 Ubuntu 基金会，还是 Google？前者已开发出无与伦比的开源操作系统，几乎终结 Linux 四分五裂的局面，后者则推出(大致上)免费的产品，有时将广告纳入其中。

Google 制造了许多很棒的产品，现在却试图在拥抱开源社群与控制自制软件之间寻求中间路线，但这是不可能的任务，肯定会惹来妄自尊大的批评，就像数十年来操作系统厂商受到的批判一般。总算觉得光明在望——一种版本集结 Linux 的精华，在众多版本中脱颖而出。我指的当然是 Ubuntu。

若你最近参加 Linux.conf.au 这类的会议，就会发现许多 Linux 笔记本型计算机用的是 Ubuntu。我在四年前就把家里用的 Linux 桌面型计算机转为 Ubuntu 版本，今年连媒体中心(Media Center)计算机也跟着换操作系统，正式挥别 Windows XP。

至少就桌面型计算机(服务器方面 Red Hat 似乎已是赢家)而言，Ubuntu 的势力崛起，大大地增强 Linux 社群挑战 Windows 和苹果平台的竞争力，因为 Ubuntu 让软件开发者能够聚焦于单一的平台，专心于软件本身的改良，这就某种程度而言是前所未有的。

评论：微软开源姿态的真正含义

近日，微软已经根据 Linux 采用的 GPL 发布了三个驱动程序，意味着将 2 万行装置驱动程序代码贡献给了开源社区。微软的这颗小炸弹引来了 IT 界的热论，微软观察家 Serdar Yegulalp 发布此文，分析了微软此举的含义。

微软对 Linux kernel 的贡献以及扩展 Moodle 的发布，确实值得我们就微软当前和未来的开源战略来谈一谈。但不论它的腿脚多么灵便，猪仍然是不会飞的。

第一：微软这么做完全是为了自己，所以别把这事儿看得太了不起。没有企业级的公司原意不顾自身利益而无私地向开源奉献。我不是在悲观，而是现实。微软向 Linux 内核做贡献，是为了人们能够更好地使用 Linux 与 Windows，从而从中受益。



而且，大多数企业都由于类似的原因而正在做类似的事情。硬件制造商发布 Linux 内核的驱动程序作为一种促进其硬件销售的方式，那么，为什么微软就不能发布一个内核模块来促进自己软件的销售呢？典型的反例是，每售出的 Windows Server 许可证就是差不多一个 Linux 支持合同的售出，或者类似的影响，只是有些媒体总是愿意说 Windows 和 Linux 不是相当量级的。

第二：即便如此，也不意味着微软现有的封闭源代码将开源，开源本就不是微软的运作方式。一直以来我总是听到一些不明就里的声音，呼吁技术界将较旧版本的 Windows 开源，我只能报以无奈地一笑。一，即使他们想他们也可能无法做到这一点；二，没有任何一方会通过此举得到利益，因而只能向前发展。如果微软曾经提供过开源的产品，在公共看来，它们将完全是从头彻尾的新项目而不是开源版本的旧东西。

第三：微软将会推出更多的开放源码，但目前这些代码主要是组件和加载项，而不是核心产品。而且，迄今为止他们个人发布的或工作在其上的大多是服务器级别的材料，而不是人们可以使用在桌面电脑上的这可能是由于开源在这个领域作出了最大的进步。他们未来的开源贡献和发布也将不会是服务器或者桌面的核心产品，而是组件。之前我已经提到，这是一种反向的开放式核心的战略：在这里，他们卖给你平台和产品，但组件和加载性等配件将开放。

第四：通过这次的开源举措，微软在以后遇到专利问题时可以轻易地面对了。上次我与 Sam Ramji 交谈，他是微软开源的倡导者，他明确表示，他已经就开源问题尽了最大的努力只是收效甚微。然后发生了 TomTom 诉讼案件（见文末注释）有人理解为这是对 Linux 的一次攻击，但在我看来是对 TomTom 公司的公司。现在，深陷于此次诉讼中，可以这么说，他们需要加倍努力，以避免类似的事情。这并不是说他们不应该保护自己的知识产权，但是他们应该花费额外的努力来避免做出类似攻击 Linux 或开源的行为。

注释：微软于 2009 年 2 月底向华盛顿的美国地区法院提出起诉，指控 TomTom 侵犯了微软的 8 项专利。作为回应，2009 年 3 月底，TomTom 向美国弗吉尼亚东区联邦地方法院对微软提出起诉，称微软 Streets and Trips 产品侵犯了与该公司汽车导航软件相关的 4 项专利。该案被视为微软起诉 Linux 的第一案。

传统路由技术的中年危机和开源新势力

在前段时间的 Interop 大会上，来自 Cisco、Juniper 和 Vyatta 的高管们针对路由技术的选择进行了一次辩论。作为传统路由技术和开源路由解决方案的典型代表，几家公司的讨论具有一定价值，以帮助用户重新考虑路由技术的选择。

关于路由技术选择的辩论



Vyatta 公司的战略副总裁 Dave Roberts 大胆挑战了当前考虑路由技术和路由器设备的思路。他认为，就定制硬件而言，今天的网络技术市场就像是 60 年代的大型机市场，由一家厂商统治当前的网络技术市场，传统路由技术厂商避免使用开源解决方案。他承认 Cisco 拥有“很好的技术”，并且这些技术在市场上存在一席之地。但是，他坚信路由功能是一种商品，可以通过开源解决方案提供，价格仅仅是传统路由器的几分之一。

Cisco 公司营销主管 Jonathan Davidson 提到, Cisco 的设备拥有开源解决方案所不具备的高可用特性, 借此回答了 Roberts 的质疑。虽然 Roberts 在这一点上没有明确表示反对 Davidson 的观点, 但他向 Davidson 询问了支持此类高可用特性的最低成本 Cisco 路由器的价格。Davidson 回答说价格至少为 35000 美元。

Juniper 公司产品线管理高级主管 Amir Khan 站到了 Davidson 一边。他说, IT 机构必须更加关注复杂和高级的路由功能, 比如路由器自我监测和支持无中断路由的能力等。

这场辩论带给我们一些思考, 如果你关注路由技术的话, 就会发现正在发生很多事情, 比如放弃专用设备的趋势。这有点像是“回到未来”, 因为第一批路由器就是运行在小型机上的基于软件的路由器。你可能会问一些问题, 现在出现的向开源方向发展的运动是合理的吗? 我们真的需要这些路由器吗? 另一方面, 你将遇到相反的问题。我们需要更可靠的路由器, 或者更大的可伸缩性吗?

重新考虑路由技术的需要

在与金融服务业交流时, 他们经常会说: “我们正在向零时延的方向发展, 现在已经出现了一些微秒级的应用。”比如, 你是个经纪人, 下单购买某支股票。那么下单的过程最好立即完成, 否则可能会由于 1 亿股股票的价格上涨 1 个点或半个点, 而付出重大的代价。

因此, 我们对于具有令人难以置信的低时延的路由技术, 的确有着真正高端的需求。与之相反。也存在一些对开源、更低成本和基本功能性的需求, 以及对难以置信的可伸缩性的需求。如果我们谈到数据中心里的成千上万台服务器, 每一台服务器上都运行着 10 个虚拟机, 必须能够在数万个不同设备中完成交换和路由, 那就必须重新考虑这种基础构件。

当 IT 机构部署新路由器和交换机时, 他们通常希望这些设备具有非常长的生命期。但是, 面临的挑战是确保这些设备不仅能支持今天的需求和近期可能出现的需求, 而且还能支持许多年之后才会出现的需求。

当前一个影响路由技术的趋势是, 多数 IT 机构将服务器从分支办事处中转移到集中式的数据中心。这些 IT 机构还在减少所支持的数据中心数量, 并在可能的地方部署服务器虚拟化。这造成了“把鸡蛋放在一只篮子里”的现象, 企业数据对于业务运营变得更加重要, 因此必须提供最高水平的可用性, 以及更高水平的传输流汇聚和可伸缩性。

一家大型金融服务机构的架构副总裁表示, 该机构部署了多种形式的虚拟化技术, 而推动这些项目的根本因素是优化成本。他补充说, 作为虚拟化的附加结果, 从前数据中心达到 5 个 9 的可用性目标变得不再被接受, 新的目标是 100% 的可用性。

目前有许多可用来提高路由器可用性的技术, 包括可在发生组件故障时无中断运行的冗余硬件子系统, 以及高可用性网络操作系统。后者支持像使用中的软件升级、冗余路由处理器间无冲击故障切换, 以及无冲击进程和协议重新启动等特性。

在讨论路由器的可伸缩性时, 必须检验当启动多种特性时性能不会受到负面影响。例如, 在整合的数据中心中, 最好启动多种安全特性(如访问控制列表、防火墙特性集合和传输流事件日志等)。经常要用到的其他路由器特性和服务还包括应用识别和 QoS 策略执行。

Vyatta 开源路由的 MPLS 实践

遵照 Linux 发行商惯例, Vyatta 分发免费版路由软件, 销售最新更新版本的软件并提供支持。在今年 3 月份的时候, Vyatta 在其最新版本的开源路由平台 Vyatta Community Edition 5(VC5)中增加些安全工具, 包括 SSL VPN、入侵检测、Web 缓存、URL 过滤和其他特性。作为免费软件,

VC5 可从 Vyatta 网站上下载。但这种免费版本不经常更新，而且不包括支持服务。商用版本的价格为 747 美元，安装这种软件的设备价格为 797 美元。

其中，Vyatta 在 VC5 中支持开源版本 SSL VPN 软件 OpenVPN。SSL 加入到 IPSec、点到点隧道协议和 2 层隧道协议的行列中，成为这种路由平台中的 VPN 选择。SSL 经常用于保护基于 Web 的交易的安全，是一种比 IPSec 等其他选择更容易与 NAT 配合使用的相对轻型的加密系统。

Vyatta 还添加了一种基于开源入侵检测系统 Snort Inline 的入侵和防御系统。这种系统可利用基于 Snort 的特征和基于网络的检测机制检测入侵企图。然后，VC5 可以抛弃与检测到的攻击相关的数据包来防御入侵。网络管理员现在可以利用 URL 过滤防止用户访问某些网站，在 Vyatta 上执行安全策略。VC5 发挥 Web 代理服务器的作用，检测 URL，阻止网站，向用户发送错误消息。VC5 还可以作为用于 Web 缓存的代理服务器，保存数据，使多台客户端不必分别通过 Internet 下载数据。新以太网绑定功能使用户可以将多条以太网链路聚集为更大的虚拟链路。

此外，其他增强特性包括用于管理基于 VC5 系统的新的基于 Web 的图形界面、用于小型企业和分支办事处的 DNS 转发和动态 DNS、提高运行 VC5 的 VMware 和 Xen 虚拟机性能的更好的驱动程序以及对更多硬件(包括 3G 移动数据调制解调器)的支持。

环境工程公司 SCS Engineers 在迁移总部与弗吉尼亚 Reston 办事处之间的 MPLS 网络时计划采用 VC5。SCS 高级系统分析师 Jerry Keene 说，作为典型工程承包商的 SCS 利润微薄，因此想方设法降低成本。不过，公司采用 Vyatta 产品的意图不仅仅是省钱，而是尽可能使用普通商用硬件，避免使用专有平台。Keene 的观点是“各种主要计算资源都将成为普通商品。”

Keene 认为，不管 SCS 未来采用什么系统，Vyatta 都应当与他们合作。他说，这种软件还在所有 Vyatta 路由平台上都是一致的，因此对于大多数网络工程师的使用比较容易，即使他们以前没有使用过开源软件。

Keene 说：“你肯定要使你的 IT 部门的生活变得更容易。你不想将自己锁定在某一家厂商上。那会是一件可怕的事情。”他说，SCS 已经在乔治亚州 Alpharetta 市的一个小办事处中部署了 Vyatta，并取得了很好的结果。

Keene 认为，VC5 中的 SSL VPN 功能将在 MPLS 网络中派上用场。SCS 8 年来一直使用 IPSec VPN。但是他说，在 MPLS 网络上利用 IPSec 完成同样的事情可能十分复杂，有时需要完成一些花费大量人力的任务，如创建静态 IPSec 隧道。但由于 SSL 更为灵活，因此实现起来也容易一些。

专家专栏

国内开源尚需耕耘 国际交流凸显重要

-----开源中国开源世界高峰论坛部分嘉宾专访

新的一期《专家专栏》又和大家见面了，本期我们有幸在开源中国开源世界高峰论坛上采访到三位国际上有名的开源人士，他们分别是 Linux 基金会执行主席 Jim Zemlin 先生、Ubuntu 社区创始人 Mark Shuttleworth 先生和 OpenOffice.org 的社区开发经理 Louis Suárez-Potts 先生。



(左起：ChinaUnix 社区编辑：周荣茂；OpenOffice 社区开发经理和社区委员会主席 Louis Suárez-Potts；Novell 中国开发经理：刘俊贤)

Jim Zemlin 先生曾任 Free Standards Group 执行董事，目前任 Linux 基金会执行董事。Zemlin 以前曾担任 Covalent Technologies 的营销副总裁，Apache web server 产品和服务的领导者。

Mark Shuttleworth 是 Ubuntu Linux 创始人、Canonical 公司 CEO，他还是一位热爱技术，创新及太空旅行的非洲企业家。Mark 是 HBD 风险基金和 Shuttleworth 基金会的创始人。2004 年初，Mark 创立了 Ubuntu 项目。

Louis Suárez-Potts 先生长期担任 OpenOffice.org 社区开发经理和社区委员会主席。他于 2007 年加入 Sun Microsystems，2000 年开始领导 OpenOffice.org 工作。Suárez-Potts 先生是 OpenOffice.org 多个项目的领导、合作领导、主要发言人和代表。

ChinaUnix 作为本次大会的合作社区，我们在大会期间内采访了以上三位开源业界著名人士，问题涉及开源业界的现状和发展，以及中国开源的发展和未来等。

ChinaUnix: 我们的第一个问题是给 Jim Zemlin 先生的，给 ChinaUnix 的网友简单地介绍下 Linux 基金会这个组织么？它是如何在业界和社区之中发挥作用的？

Jim Zemlin: Linux 基金会主要负责 Linux kernel 开发，包括雇佣 Linux 的核心人员，比如 Linux 创始人 Linus Torvalds。另外，我们还负责 Kernel.org 网站的维护和更新工作。以上可以说是 Linux 基金会的最重要工作。目前 Linux 基金会由很多企业成员，比如红帽、Novell、Intel、IBM 和 HP 等，这些企业的加入可以保证 Linux 的发展更加具有中立性，而不会偏向于某些企业或者趋势，另外，他们

自身的利益也能够得到保证。同样我们也有个人会员，但是需要缴纳少量的会费。

除了内核的发展和监护外，Linux 基金会还负责另外一些开源项目的发展和监护，比如最近刚刚加入的 Moblin 项目，Moblin 项目的目标是手持设备和上网本等新兴设备市场。

另外，Linux 基金会还有以下一些功能：

- 1、支持和协助一些硬件厂商开发和开放他们的硬件驱动；
- 2、为 Linux 和开源软件提供一些法律方面的支持，包括辩护和专利案咨询等等；
- 3、针对 Linux 和开源软件行业和社区进行一些培训、教育和推广活动；
- 4、维护 LSB 项目的发展，保证 Linux 平台的兼容性和发展一致性；
- 5、维护 Linux.com 网站，提供包括资讯在内的一系列 Linux 内容给所有人。

ChinaUnix: 现在业界的新概念中，比如上网本、云计算中，Linux 出现的频率都非常高，我想问一下，Linux 将会在这些新平台或者新应用中扮演什么角色？

jim Zemlin: 首先，大家都有看见，Linux 现在是一个非常成熟的操作系统，不仅仅是在服务器操作系统市场，在桌面 Linux 市场，嵌入式平台和手持设备系统，都有大量的 Linux 系统在投入使用。在新兴的技术平台也是一样，Linux 将会扮演非常重要的角色。比如在超级计算机市场，我们看到，在 Top500 排行中，Linux 占据了 90% 的市场。

在云计算平台，我想 Linux 会得到很好的发展，因为 Linux 更具有基础性和开放性，比如在云安全、虚拟化方面，具有开放性的 Linux 占有了先机。

ChinaUnix: 大家都知道 Linux.com 这个网站，现在已经属于 Linux 基金会，它将在 Linux 业界和用户社区中发挥什么作用？并且听说将会推出中文版，那么 Linux.com 在中国将如何推广？

jim Zemlin: Linux 基金会下属的 Linux.com 网站，可是算是一个全新的 Linux 内容网站。现在大家不缺少 Linux 资讯和技术内容，你可以通过搜索引擎得到大量的这方面的信息，但是你没有办法分清主次，这是问题的关键，Linux.com 的重新推出就是为了解决这个问题的。先是通过我们的组织，然后再是用户的选择，Linux.com 将会为更多的用户产生主次分明的，能够指导用户或者企业发展的技术和资讯内容。

同时，Linux.com 将会是一个大的平台，这里面有标准制定者、厂商、用户、ISV 等等，还有社区志愿者，他们是不可忽视的一部分。作为一个大的交流平台，我想 Linux.com 将会发挥一个平台的作用，很好地将多方面的力量融合起来，共同推动 Linux 朝着良性的方向发展。

至于推出中文版的事情，确有其事，因为将来 Linux.com 会在很多地方得到发展，不光是中国，还有很多别的国家，包括发达国家和发展中国家，因为母语是最适合当地技术人员交流的语言。Linux 基金会将不仅仅是将英文内容翻译成中文，传播给中国的用户。因为大家都知道，国内很多的技术文档和资料不是很丰富，没有国外的及时，这确实是个问题。但是大家也应该看到，其实国外也不知道中国国内 Linux 相关行业和社区的发展情况，特别是国内 Linux 市场、技术发展和社区规模等方面的情况。这就需要国内相关的人员和企业，还有政府通力合作，将国内的相关信息翻译并传播到欧美和国际开源社区。只有做到互通有无，国内和国际加深了解才能更好地促进 Linux 和开源技术在中国的应用和发展。

我想 Linux.com 将在上面所提到的问题中起到一个积极的交流平台作用。

ChinaUnix: Linux 桌面市场一直发展的非常缓慢，到目前为止，占有率还停留在 1% 左右，我想知

道，Linux 基金会是否在企业会员的影响下，从计划上就对桌面有所忽视，比如在内核开发和推广方面，服务器相关方面的内容要占有优先地位？

Jim Zemlin: Linux 基金会没有偏心，没有说服务器相关方面的投入要大于桌面方面。其实我们的看法需要改变，因为传统的桌面在发生改变，桌面和移动的分界线正在逐渐模糊。我想在不久的将来，突破 1% 的市场份额还是很有希望的。首先，Linux 操作系统的成本低，在目前的桌面 PC 电脑和上网本越来越低的价格面前，价格是桌面市场的主要因素，微软和苹果的产品劣势明显，因为他们不可能有 Linux 那么便宜；其次，Moblin、Android 和其他一些目标在 MID 和移动平台的基于 Linux 的操作系统也是占据了新的制高点，他们可定制性强，生态系统完善等等，这些都注定了他们将来会在更广范围的桌面操作系统市场占据更多的份额。

ChinaUnix: 好的，接下来的问题是我们为 Mark 准备的，Ubuntu Linux 目前发展势头良好，在中国将做何打算，现在是否已经有打算，将来会如何发展？

Mark Shuttleworth: Ubuntu 在中国主要有两方面的工作要做。首先，中国有着大量的 Ubuntu 和其他 Linux 用户，他们对自由软件、Linux 和其他开源软件抱有很高的热情，这确实很令人高兴，我们也将继续支持 ubuntu Linux 在国内开源软件社区的活动，继续在社区中推广 Linux 和开源软件。将国内开源软件社区和全球的开发者和用户联系在一起，促进大家更好的交流。

其次，我们也看到，在国内的个人电脑市场，越来越多的个人电脑开始预装 Ubuntu Linux 操作系统，这也是我们在国内的一个重要工作。其实我们已经在这方面做了大量的工作，特别是现在上网本市场的崛起，使得我们看见 Linux 的灵活性非常适合上网本等新兴的硬件平台，也有不少的硬件厂商开始和我们合作，帮助测试 Ubuntu Linux 在这些新平台上的兼容性和性能，我很高兴看到在中国有这样的 ubuntu Linux 商业生态系统逐步地得到建立。你也知道，Canonical 在和 OEM 以及计算机操作系统预装这块也有不少成功的经验，比如在美国市场和 Dell、HP 的合作等等。我们也正逐步地将这些成功的经验移植到中国市场上来。我们也很希望在不久的将来，能够看到中国有更多的软硬件厂商能够提供更多的基于 Ubuntu Linux 的解决方案，我也很乐意看到。

ChinaUnix: Ubuntu Linux 在桌面 Linux 市场表现的很好，在服务器市场将如何拓展？

Mark Shuttleworth: 我想说的是，在桌面 Linux 市场，易用性可能是 Linux 发行版最关注的。我们知道，现在有很多的 Linux 发行版，他们有着一群特定的用户群体，因为他们在某个方面都做的非常有特色，就像你说到的那样，Ubuntu 在 Linux 桌面市场赢得他应该有的尊重。

那么在服务器操作系统市场，我想 Ubuntu Linux 应该有不同于其桌面市场所遵循的原则。Ubuntu 在桌面操作系统市场注重用户的易用性，但是在服务器操作系统市场，Ubuntu Linux 可能遵循的是易部署性和高效性，提供更多的企业级应用程序、提供更加专业的技术支持等等。我们现在也有看到，Ubuntu 也已经在一些主机托管公司的数据中心得到部署，这些数据中心运行着成千上万台的服务器，为很多中小企业提供主机托管业务。虽然现在 Ubuntu 还没有提供专门为企业定制的服务器 Linux 操作系统，但是我们现在也开始提供通用版本的服务器 Linux 操作系统，他们拥有更多的可定制性，满足大规模环境部署的需求。在不久的将来你会看见，Ubuntu 的服务器操作系统将会更多的注重平台的基础性，以及提供更优越的操作系统性能。

ChinaUnix: Ubuntu 项目的背后母公司 Canonical 目前盈利么？Ubuntu 和 Canonical 目前是一个什么样的关系，以后将如何发展？

Mark Shuttleworth: 首先，Canonical 是一个以盈利为目的的商业公司，提供 Ubuntu 的支持服务和解决方案，Ubuntu 是一个开源项目，两者是相互独立的。但是，Canonical 对 Ubuntu 项目提

供了大量的支持工作，包括 Canonical 的很多员工都是在为 Ubuntu 项目做贡献，从这个角度来说，两者又是密不可分的。Canonical 公司目前的商业模式主要是基于软件的服务，提供基于 Ubuntu 和其他开源软件的解决方案。Canonical 目前还没有盈利，尚不能收支平衡。Canonical 公司依赖软件的服务盈利，这是现在，也是未来一段时间内开源软件公司的发展趋势。我们想将来能够有更多的企业用户或者付费用户能够来采用我们的开发平台和软件服务，这也正是我们所追求的，我想这才是 Canonical 将来的盈利方向。Ubuntu 项目，大家都知道，除了 Canonical 外，还有很多公司和志愿者参与其中，这是一个社区项目，在这个社区中的公司和个人，都是有着热情和理想，那就是想把 Ubuntu Linux 做好，能够得到更大范围的用户的的支持。唯一不同的是，Canonical 可能在这个项目中拥有更大的影响力。

ChinaUnix: 最后我们要采访的是 Louis 先生，能简单的介绍一下 OpenOffice 项目以及目前的发展现状么？

Louis Suárez-Potts: OpenOffice 是一套完整的桌面办公软件，目前处于 3.0 版本的发行周期，目前有超过百万的用户在使用 OpenOffice 办公套件。OpenOffice 支持多个操作系统平台，包括 Linux、Solaris 和微软的 Windows。经过这么多年的发展，OpenOffice 办公软件所支持的平台以及兼容性已经做的很好，特别是跨平台，以及和微软 Office 办公软件的文档格式交互方面。所以你有看到，越来越多国家政府开始使用 OpenOffice，而且他们推荐 OpenOffice 成为政府和其他国家机关的日常办公软件。另外，因为 OpenOffice 使用的文档格式是 ODF，这样就使得他们不会被某些厂商所劫持，只能使用一种或者某些特定的文档格式，不能随便的转换到其他文档格式，这对于一个政府和国家来说，确实很可怕，大家都不想这种事情发生。

ChinaUnix: 能简单的说一下 Sun 和 OpenOffice.org 项目的关系么？还有，在 Oracle 收购 Sun 之后，OpenOffice.org 项目将如何发展？

Louis Suárez-Potts: 哈哈，我还是先回答你的第二个问题吧，你也应该知道，Oracle 在英文里的意思是“占卜的人”，能够获得神的旨意预知未来的人，但是很不幸，我不是 Oracle，我不能预知未来。但是我们可以看到的是，OpenOffice.org 不仅仅是由 Sun 来支持的，Sun 只是其中最主要的贡献者。而且，到目前为止，Oracle 和 Sun 所发表的公开声明中，没有提到要伤害 OpenOffice.org 项目存在和发展的举措，因为这对两家公司都没有好处。从目前来看，OpenOffice 的用户数量不断快速增长，从授权以及项目的发展目标来说，Oracle 对 Sun 的收购对其基本不会产生影响。

ChinaUnix: 您也应该知道，处于国家安全的考虑，中国现在也在制造一些信息技术方面的标准，比如 3G 通信技术、局域网传输协议和文档格式等，像 OpenOffice 这样的软件追求的全球统一的文档标准，OpenOffice.org 在遭遇到这些国家标准壁垒的时候，是如何处理的？

Louis Suárez-Potts: 我首先想举个例子，南非目前就在使用 OpenOffice 作为他们政府的通用办公软件平台，他们最关注的是易用性，这个易用性包括文档格式兼容、安全和高效等等。目前的 OpenOffice 的数字签名、文档加密等功能也做的比较完善；另外，OpenOffice 的本地化工作也做的非常好，很多地方语言都已经有了支持，在多民族多语言的国家非常得到欢迎；第三点，就是 OpenOffice 的跨平台性，因为 OpenOffice 是跨平台的，兼容性非常好，能够在多个平台实现无缝的兼容，这是目前其他专有 Office 办公软件所没有办法做到的。

总的来说，OpenOffice 很灵活，因为巨大的用户基础，使得我们有实力和其他标准坐下来谈判，获得他们的支持；同样，OpenOffice 也能够 ODF 之外，选择其他的文档格式，包括专有的文档格式。所以对于中国标准的 UOF 文档格式，我们也能够很好的支持，只要大家有合作的意向。

ChinaUnix: 作为一个社区的领导，和一个真实公司中的项目经理相比，有何不同？社区负责人应该如何来做？

Louis Suárez-Potts: 首先，我没有在很多大公司的工作经验，在 Sun 公司任职之前，我一直在教育行业工作，对项目经理这些职务不是很熟悉。但是在开源领域方面还是有一些为人处事的经验。首先，在社区里，这里没有老板没有上司，大家相互之间的尊重可能相对来说要少一点。特别是有争论的时候，你需要很好的理由来说服对方，可能在公司里不会这样，老板让你做什么你可能会照办。还有，因为网络的隐匿性，很多社区里人会很粗鲁，经常在社区里因为理念不合而相互谩骂，你作为一个社区管理人员，你得做好相应的准备，应付这些一般在日常办公室环境中所不会遇到的尴尬情况。因为在日常办公环境中，很少双方会谩骂或者大打出手的，而在社区中这些时常发生，甚至会导致社区会员分裂，成立新的社区分支情况。

但是社区领袖和项目经理的存在价值是一样的，那就是想把一个项目做好，做成功。同样在这个理念下，使用的部分技术手段也是相同的，比如通信技术、bug 跟踪和除臭系统、源码同步系统，使用论坛或者在线会议系统等等。

还有一点，因为开源项目的特殊性，开源项目对参与其中的人员保持有尊敬之意。参与其中你有一种成就感，而不像商业软件公司，你总是感觉你是一个螺丝，永远做着机械的没有创意的工作。虽然社区成员之间时有谩骂，但是这是因为各自理念差异而导致的，他们对社区的贡献我们还是要感谢的。

ChinaUnix: 好的，谢谢各位对我们 ChinaUnix 的支持，接受我们的采访，我们将一如既往地发挥好社区的作用，将业内最新的看法和发展现状传递给国内外的开源从业人员和爱好者。

本期推荐

生于有效创新 死于盲目仿制

ChinaUnix 网友: dijos

中国是否需要国产操作系统，这已经不是问题了，问题是，如何才能拥有自己的操作系统？

仿制，前路茫茫

学习→仿制→消化→超越，这向来是后来者跟踪并超越先进的捷径，然而，这种方法在通用操作系统（PC 或嵌入式）上却行不通，为什么呢？

ucosii 的作者 Labrosse 先生在其书中说“写个实时内核那么难吗？不就是不断地保存、恢复 CPU 的那些寄存器嘛”。这句话并不全面，而且误导了不少人，他们引而伸之地认为：“写个操作系统那么难吗？不就是实现线程和进程管理、信号量管理等 API 函数嘛”。事实如何呢？写个成功的操作系统真的就那么简单吗？我们来看看一个典型、有一定复杂度的软件开发过程，如图 1 所示。

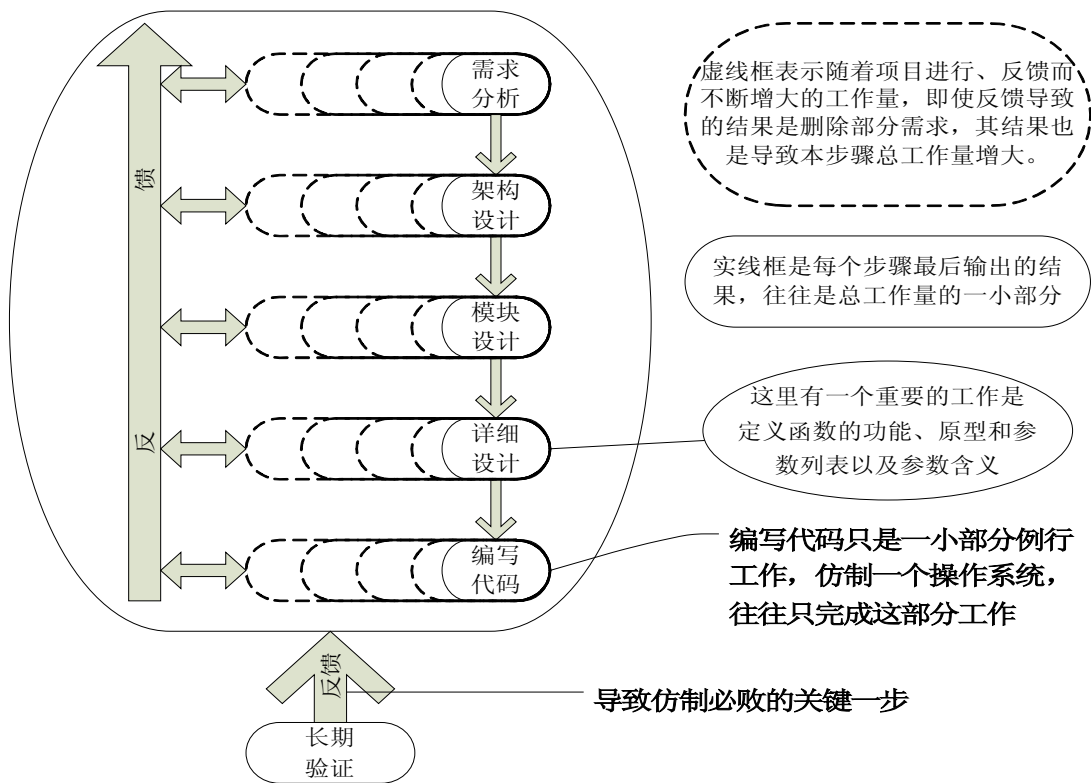


图 1

什么是仿制呢？用类似的底层算法，实现一套功能大同小异的 API 函数，即使每一行代码都是你自己写的，也叫仿制。

常听到有人说，现在写一个操作系统比 20 年前容易多了，这句话实际指的是仿制一个操作系统。容易在哪里呢？现在有发达的网络，漫天飞舞的开源资料可以借鉴，而 20 年前，当 linus 先生仿制 UNIX 完成 linux 的时候，或者 Labrosse 先生仿制 RTOS 制作 ucosii 的时候，是没什么开源资料可参

考的，被仿制的对象几乎是唯一的参考资料。而今天，人家连函数原型都已经定义好了，又有那么多参考资料，虚拟机、开发板这些软硬件开发平台又是那么容易得到，难怪现在 3 个月就能写出一个不错的内核出来了，甚至连一些中学生都跃跃欲试。我们目前看到的漫天飞舞的操作系统中，绝大多数就是只完成“编写代码”这一小部分工作的仿制品。

所以，Labrosse 先生的话应该改为“仿制个实时内核那么难吗？不就是不断地保存、恢复 CPU 的那些寄存器嘛”。引而伸之地可以说，“仿制一个操作系统那么难吗？不就是实现线程和进程管理、信号量管理等 API 函数嘛”。然而，仿制只能仿一个外形，并不能复制他们在市场上的成功，操作系统业界多年没有成功的新苗冒出来了，也是这个原因。

为什么说仿制必败呢？不可否定，新编写的操作系统，在前人的基础上，可能可以写出更精巧的上下文切换函数、更高效的 API 函数、更低的资源消耗、更快的中断响应，但是，一个操作系统的质量并不完全由这些因素构成，数十年、海量用户的使用和反馈为基础的稳定性和可靠性，以及用户长期积累的信心才是根本。如果一个新出的操作系统，只是实现原有操作系统类似的功能，你到哪里捕捉你的第一只螃蟹？即使有了第一只，那第二只、第三只呢？

所以说，如果没有根本性的创新，仿制一个操作系统是死路一条，即使以谷歌的实力，也只是在 Linux 基础上开发 Adroid，而不是新仿制一个操作系统。

信心，源自有效创新

都江堰操作系统（下文简称 djyos），是一个全新的操作系统，它在操作系统的几个重要方面：调度器、中断管理、设备管理、资源管理等方面，都又非常突出的创新，甚至是理论创新。djyos 花了 5 年时间设计，才初具雏形，因为它完成的是如图 1 所示全过程的工作全新操作系统，这 5 年间，经过无数次的反馈，方案被无数次推翻，许多策略被多次重新定义，许多代码被多次重写，才有你所见到的 2 万行代码。而不是像一些仿制的操作系统一样，调度策略、api 设置都是享用现成的操作系统策略，只完成“代码编写”部分的工作。由于 djyos 在构成操作系统的多数主要方面都跟现有操作系统有着巨大的不同，因此很难参考现有系统的设计，反而为了在创新的同时尽可能照顾现有习惯而不得不有所顾忌。djyos 的创新主要体现在以下方面，这些都是传统操作系统没有的。

(1) 在调度器方面，可以说是彻底革新，它几乎颠覆了传统的软件设计模式。对嵌入式编程来说，几乎是革命性的进步，它以事件为核心的调度方式，改变了过去操作系统要求程序员模拟计算机的思维方式的编程模式，使之可以按人类的思维方式编写应用程序；它使程序员可以摆脱线程和进程的束缚，djyos 没有跟线程相关的 API。特别是在嵌入式领域的程序员，有许多是行业背景很深的专家，但往往缺乏计算机操作系统方面的知识，他们在 djyos 下编程，则可以无需学习晦涩难懂的线程知识。

(2) 天然匹配多核和分布式计算，djyos 的调度算法是以事件为核心的，应用程序只管为事件编制处理程序，而线程和 CPU 核则被当做处理事件必须的资源，由操作系统分配给事件，这种算法天生使多核编程更加简单，多核利用率更高。众多周知，目前 CPU 正往多核化发展，嵌入式 CPU 甚至 DSP 也是如此，比如 ADI 公司的 blackfin561 就一颗只有几十 KB 内存的双核 DSP，djyos 有计划移植到 blackfin561 上，有望成为第一个在只有几十 KB 内存的 CPU 上实现对双核的完美支持的 RTOS。

(3) 创新的加载方式和中断管理模块，可以实现比所有操作系统都快中断响应时间，并且在 CPU 启动后数十 μs 以内，应用程序的关键部分就可以实现对目标系统的控制，在操作系统加载过程中，就可以响应应用程序的中断 ISR。

(4) 实时的内存管理，在操作系统组件中，通常认为 malloc 函数是非实时组件，因为即使有充足的内存，malloc 函数的执行时间也不可预计，不符合实时系统要求执行时间和结果均可预计的原则。但是 djyos 的内存管理，只要在系统设计时配备充足的内存，malloc 的执行时间是快速而且可以预计的，而在只运行一个应用程序的实时系统中，内存不枯竭是容易做到的。

(5) 为项目经理和系统工程师量身定制的系统架构，为工程师量身定制的 API 设置，特别利于系统工程师划分项目模块，利于项目经理组建和管理团队，利于工程师编写利于维护的代码。

(6) 欲了解更多的特长，请到 ChinaUnix 项目孵化专区下载《都江堰操作系统与嵌入式系统设计》一书。

价值，来自用户

仿制必败，并不意味着创新必胜，创新只是必要条件而非充分条件，只有能为用户创造价值，才是充分条件。djyos 的创新优势在于，它不是为了创新而创新，它的每一个都是为了给用户创造价值。我们来看看操作系统界的几个“怪”现象：

- 1、互联网上搜索一下 wince 和 linux 的技术指标对比，几乎 linux 完胜。
- 2、来自 Gartner 公司的数据证实，自从 2003 年起微软就一直稳居嵌入式操作系统全球市场份额榜首。
- 3、嵌入式 linux 强调免费且硬件成本低的优势，而 wince 最成功的地方，恰恰是成本最为敏感的、出货量巨大的消费电子产品。
- 4、《电子产品世界》2008 年在工程师中做的调查表明，开发 linux 应用程序的工程师数量是 wince 的 2.5 倍。（注：linux 工程师的平均工资还远高于 wince 工程师）

这些数据表明，linux 能在技术指标上完胜 wince，说明 linux 是有了不起的创新的，但是，linux 以多得多的成本，所开发的产品市场占有率却远比 wince 低！不能不说，linux 的创新知识 linuxer 们自娱自乐的玩具而已！应用这些创新能创造的价值，需要消耗太多的人力去学习 and 掌握 linux，这些消耗足以抵消创新的价值。

所以说，linux 有巨大的创新，却大多是无效创新，这些创新使得高级专家们倍加推崇 linux，专家的意见使许多项目盲目上马选择 linux，把企业置于巨大的技术风险中，成就了大量高薪的 linux 工程师，企业却在消耗了大量研发经费和时间后，离成功仍然遥遥无期。这就是 linux 从业者众但市场占有率低的原因，前车之鉴，djyos 需要防止重蹈覆辙。

是什么使人愿意花高价购买 wince 的 license，并付出更高的硬件成本呢？wince 之所以取胜，在很大程度上来自快速方便的开发，快速推出产品，技术风险小，人力成本低。在 windows 系统衍生工具的支持下，图形化的开发，事件触发式编程令人倍感方便。与 wince 通过应用程序（对操作系统来说，EVC 等开发工具属于应用程序）间接实现事件出发式编程相比，djyos 的事件触发确实原生的，即 djyos 调度器就是按照事件进行调度的。所以，你只要有一个支持 ansi C 的编译器，只要会 C 语言，再加一个文本编辑器，就可以实现事件触发式编程。原生式的事件触发编程，从理论上，其执行效率和代码尺寸也应该比用应用程序间接实现的高。

djyos 的核心算法不但在通用嵌入式操作系统中有优势，此外，djyos 还是一个 RTOS，除跟 wince 在手持设备上竞争外，还可以与 VxWorks 等实时系统竞争，在 RTOS 领域，djyos 在研发方便性上的优势更加突出。在性能上，按 djyos “九九加一”原则设计的中断管理架构，使部分中断可以永不禁止状态，从而实现最快的中断响应延迟。我们知道，实时系统中，实时性要求最高的部分

功能，往往是用中断实现的，最快的中断响应速度，意味着 djyos 具有最高的实时性。

成功，扬长避短

在软件领域，革命性的创新的另一个拦路虎是惯性，作为革新的操作系统，你将很难做到与现有应用程序的兼容，你将立即面对行业惯性的压力，这种惯性来自于用户以及工程师的使用习惯和应用程序的兼容程度。如果在桌面系统上，这个压力尤其显著，移动设备次之，工业产品却几乎不受这个惯性影响。诚如我在《都江堰操作系统与嵌入式系统设计》一书中指出，开发高可靠的产品，对使用外来代码会非常谨慎，而且，由于这些产品的特异性大，甚至很多是定制产品，直接可用的公共代码资源很少，即使有，也不是应用程序的形式提供，而是库函数的方式提供。再者，高可靠性工业控制产品中，一般只运行一个程序，而这个程序的绝大多数代码，都是产品研发组自己开发的，对运行外来应用程序的需求几乎没有。所以，在工业领域，操作系统只要把常用的公共软件模块如网络模块、gui 模块做好了，就基本满足用户需求了。嵌入式操作系统是一个细分市场，所以不会形成桌面系统寡头控制的局面，在桌面系统上，是 windows 和 unix 系操作系统双寡头局面，这种情况绝不会出现在嵌入式领域。这就给了后起之秀一个机会，只要你的产品好，工程师认同，你就有机会。嵌入式操作系统的目标用户是工程师，而工程师是一个善于学习的群体，容易接受新事物的群体，这也是 djyos 的机会。所以，djyos 扬长避短，避开在通用计算机/服务器上与 windows、linux、unix 正面交锋，首先寻求在嵌入式领域寻找突破点，djyos 要与非实时操作系统如嵌入式 linux、wince 竞争，以及实时操作系统如 vxworks、qnx 等竞争。djyos 提供三种运行模式，si 模式是为资源非常匮乏的单片机准备的，dlsp 模式是为资源相对丰富的中大型工控系统准备的，mp 模式是为移动计算设备准备的，djyos 将首先从工控领域取得突破。djyos 先天优异基因，将为它带来机会。

开源，加速推广

一个全新的操作系统写出来了，要经过让人们接触、了解、熟悉、接受、使用的过程，这是走向成功的必经之路，在这个成熟的市场格局之下，这点尤为重要。djyos 通过以下方法实现。

- (1) 通过宣传使人们能够接触到 djyos。
- (2) 公开全部充满注释的源代码，以及完善的文档，让人们能够了解和熟悉 djyos。
- (3) djyos 本身优秀的设计理念内涵，以及高可靠性的设计，使人们能够接受并扩大传播。

嵌入式软件的规模越来越大，越来越复杂，软件中潜藏的 bugs 也会越来越多，隐藏得越来越深，需要源代码来协助定位 bug 的需求也就越来越迫切。2005 年以来，Windows CE、VxWorks、QNX 都相继开源，有些是向公众开源，有些是向客户开源。我想，他们开源的初衷，并不是想像 Linux 那样，让公众修改他们的代码，而是市场压迫下不得不开源。所以，我认为，操作系统尤其是嵌入式操作系统，不管是免费的还是收费的，不开源就是没有出路的。对于自主操作系统来说，开源还有更重要的意义，可以藉此消除行业质疑，是证明你自主研发的最有力证据。

然而，开源后呢？一个新系统，摆一堆源代码在那里，然后你说你的系统有多么多么的先进，会有人看吗，会有人相信吗？用户把几万行代码拿到手里，根本就云里雾里。你必须有说明文档，这些文档必须详细，必须有良好的组织，必须易于阅读。djyos 的 5 年研发过程中，有 2 年在写配套的书《都江堰操作系统与嵌入式系统设计》，该书并不是简单地作为 djyos 的说明文档，还融汇了许多软硬件系统架构设计方面的思想，而这些思想又构成了 djyos 设计的基石。可以说，光这份文档的制作时间，就超过仿制一个普通操作系统内核所需要的时间。

更多本项目信息，请参考开源项目孵化专区：<http://linux.chinaunix.net/bbs/forum-74-1.html>

技术新知

独辟蹊径之 Linux 主机监控工具 munin monit ntop

ChinaUnix 网友: kns1024wh

本文介绍在 centos 5.x 环境下通过 yum 源的扩展使用 munin、monit、ntop 工具来监管你的应用程序和服务。题为懒人说说的是简便的安装方式而已，将强大的功能配置简单的应用起来是很重要的，可以节省时间并提高效率。

Monit: <http://www.tildeslash.com/monit/>

Munin: <http://munin.projects.linpro.no/>

Ntop: <http://www.ntop.org/>

当你的 Linux 主机投入生产环境以后，想要知道系统或者服务每天或者每周的运行情况，Munin 是最佳选择。在某些情况下，监控 Linux 主机服务、应用程序的运行状态，是非常有必要的，尤其是应用程序或服务会出现错误，并可能开始吞噬你主机的内存和 cpu，这将导致主机上的应用不能正常的运行。因此，需要其他一些程序来监控你的程序，在程序出错以后重启你的程序。在 linux 下解决方案，如 Nagios、ganglia、HyperHQ、ZABBIX 等，nagios 的插件很多功能也是很强大的，需要详尽的配置实现；HyperHQ 会开启 JAVA 的应用并对 JAVA 的应用服务器会自动设备，想起来 JAVA 对内存的强烈需求还是会有过估计。这这了推荐使用 Monit、Munin 以及 ntop 一个简单的网络流量的监控工具。相对前面列举的强大功能需要复杂配置，本文将描述的 Monit、Munin、Ntop 只是需要使用 yum 进行安装就可以方便的使用监控主机上的资源并发出警告邮件。

Munin 是一个类似 cacti 的性能展示工具，走的是更可靠的 TCP 协议。不会出现 cacti 基于 snmp 这样的监控方式会出现各种各样的问题。Munin 是用 Perl 编写的,有很多插件功能，支持 Unix 操作系统的多种版本，用到很多不同的 perl 包；Munin 和 MRTG 类似，也可以通过各种插件将服务器的各方面状态以图像的形式表现出来。munin 画图也是使用 rrdtool，因此 Net::Server perl 模块和 rrdtool 都已经安装在了服务器上这些必须安装 munin 前安装完毕。

Munin 的主要优点就是它能够同时提供收集信息和分析信息的功能，这样你就无需在每个节点上安装一个 web 服务器，并且详细信息页面底部的附加信息能够帮助你获取一些重要数据的统计信息。

Debian/Ubuntu 和 Fedora 都附有 munin 安装包，不过 RHEL 和 Centos 默认没有包括 munin 的安装包。

Munin 功能主要为收集和分析功能，这可以帮助你收集多个服务器信息并在一个中央服务器上分析所有收集到的信息，通过 apt 源或者 yum 源使其安装和更新都相当简单。

当你访问 <http://localhost/munin> 时，Munin 会显示一张概览图向你展示所有连接到的节点，包括到节点的具体功能，如磁盘、网络、NFS 和进程。点击节点名称会显示两行的图表，每一行都会在左边显示每日统计信息，右边显示每周统计信息，点击这些图表能够提供显示某日、某月的详细信息页面。在这项细节信息页面的底部，显示的是关于数据的更加详细的信息，包括不规则活动

通知等。

Monit 是一款运行于应用层功能非常丰富的用于对系统中的进程、文件、目录、以及设备等进行监视和管理的工具。Monit 包含一个内嵌的 HTTPS Web 界面，可以使用浏览器方便地查看 Monit 所监视的服务器。Monit 对管理员来说可谓神器也。Monit 可以自动修复那些已经停止运作的程序，特使适合处理那些由于多种原因导致的软件错误。Nagios 也就是在监控，而 Monit 不但本地监控，远程服务监控。可以做到服务就永远都能“死而复生”，是服务一直在线。Monit 可以监控的主机信息如下：

- * System Services

- o Cron (program timer)
- o Gdm (gnome desktop manager)
- o Inetd (internet service manager)
- o Syslogd (system logfile daemon)
- o Xfs (X font server)
- o YPBind (Yellow page bind daemon)
- o Net-SNMP (SNMP agent)
- o NTP (time server)
- o Nscd (name service caching daemon)

- * Name Services

- o Bind (chrooted)

- * FTP Services

- o vsftpd

- * Login Services

- o SSHD

- * WWW Services

- o Apache (web server)
- o Zope (appication server)
- o Squid (http/ftp proxy)
- o Privoxy (spamfilter proxy)

- * Mail Services

- o Postfix (mail server)
- o sendmail (mail server)
- o Qpopper (pop3 server)

- o Spamassassin daemon (spam scan daemon)
- o Amavis-new (mail virus scanner)
- o Policyd (Postfix access policy delegation daemon)
- * Virus Scanner
 - o Sophie (virus scan daemon)
 - o Trophie (virus scan daemon)
 - o Clamavd (virus scan daemon)
- * Printing Services
 - o LPRng (printer daemon)
- * Database Services
 - o MySQL Server
 - o OpenLDAP Server
- * File Services
 - o Samba (windows file/domain server)
- * Sun ONE Services
 - o iPlanetDirectoryServer (Sun ONE)
 - o iPlanetMessagingServer processes (Sun ONE)
 - o iPlanetCalendarServer processes (Sun ONE)
- * Misc Services
 - o apcupsd (APC ups daemon)
 - o Webmin (remote admin service)
 - o STunnel (SSL tunnel)
- * Misc Usage
 - o Watch and analyze crashdumps (Solaris)
 - o Watch and analyze crashdumps (Linux)
 - o Start and stop tcpdump based on condition
 - o Rotate tcpdump until condition occurs
 - o MySQL event driven process list

Ntop 跟 top 监视系统活动状况相似，ntop 是一个用来实时监控网络使用情况的工具。由于

ntop 具有 Web 界面模式，因此无论是配置还是使用都很容易在短时间之内快速上手。Ntop 让网络流量透明化，NTOP 显示网络的使用情况比 MRTG 更加直观、详细，NTOP 甚至可以列出每个节点计算机的网络带宽利用率。NTOP 是一个灵活的、功能齐全的，用来监控和解决局域网问题的工具。它同时提供命令行输入和 WEB 界面，可应用于嵌入式 WEB 服务。可以通过分析网络流量来确定网络上存在的各种问题；也可以用来判断是否有黑客正在攻击网络系统；还可以很方便地显示出特定的网络协议、占用大量带宽的主机、各次通信的目标主机、数据包的发送时间、传递数据包的延时等详细信息。通过了解这些信息，网管员可以对故障做出及时的响应，对网络进行相应的优化调整，以保证网络运行的效率和安全。

上面介绍的 munin, monit, ntop 工具传统的基于源码的安装方式相对要解决的软件包的依赖关系比较多，这个也是很多 Linux 的管理员选择监控工具相对比较单一的安装因素。很多人都是习惯于使用 redhat 的发行版本，尽管 yum 是在软件包安装方面比较的强大不过默认的 redhat 的官方的 yum 源和 Centos 的社区源是没有 munin, monit, ntop 这三个软件包源提供的，而在 Ubuntu/Debian 的 apt 的源中默认都是有支持的。本文将告诉大家如何使用第三方的 RPMforge Repository yum 源来快速安装 munin, monit, ntop 工具并简单的配置，实现 Linux 主机网络的监控的应用。

RPMforge Repository 源分为 i386 和 X86_64 位版本

在 i386 版本在中的安装方式

```
wget http:#packages.sw.be/rpmforge-release/rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

```
rpm -Uvh rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

```
rm -f rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

在 x86_64 版本中的安装方式

```
wget http:#packages.sw.be/rpmforge-release/rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
```

```
rpm -Uvh rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
```

```
rm -f rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
```

(<https://rpmrepo.org/RPMforge/Using> 最新版本信息).

安装后在/etc/yum.repos.d 文件夹中生成 mirrors-rpmforge rpmforge.repo 配置信息。

有了这个 RPMforge Repository 的源就可以轻松的开始 munin, monit, ntop 的监控之旅。

安装和配置 munin

执行 yum 完成安装

```
yum install munin munin-node httpd
```

如果没有安装 apahce 需要添加上 httpd 参数

设置 munin 启动脚步并启动 munin

```
chkconfig --levels 235 munin-node on
```

```
/etc/init.d/munin-node start
```

简单设置 munin 的配置文件/etc/munin/munin.conf

```
vi /etc/munin/munin.conf
```

```
dbdir /var/lib/munin
```

```
htmldir /var/www/html/monitoring      #此行设置 web 页面及图  
形数据的存放位置
```

```
logdir /var/log/munin
```

```
rundir /var/run/munin
```

```
# Where to look for the HTML templates
```

```
tmpldir /etc/munin/templates
```

```
# a simple host tree
```

```
[localhost]
```

 #此部分定义一个监控主机的信息

```
    address 127.0.0.1
```

```
    use_node_name yes
```

新建/var/www/html/monitoring 文件夹并设置权限

```
mkdir -p /var/www/html/monitoring
```

```
chown munin:munin /var/www/html/monitoring
```

```
/etc/init.d/munin-node restart
```

观察/var/www/html/monitoring 的文件夹的信息如下

```
[root@server146 monitoring]# ls -l /var/www/html/monitoring
```

总计 28

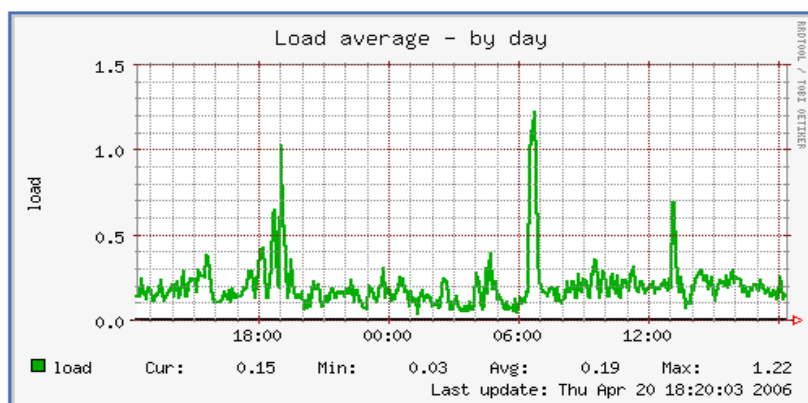
```
-rw-r--r-- 1 munin munin 2555 07-22 15:15 definitions.html
```

```
-rw-r--r-- 1 munin munin 1584 07-23 17:00 index.html
```

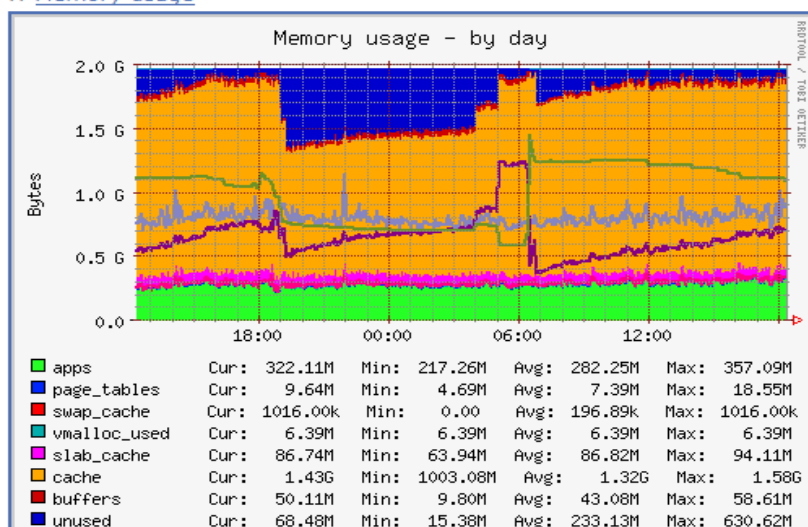
```
drwxr-xr-x 2 munin munin 12288 07-22 15:15 localhost
```

```
-rw-r--r-- 1 munin munin 473 07-22 15:15 logo.png
```

```
-rw-r--r-- 1 munin munin 3538 07-22 15:15 style.css
```

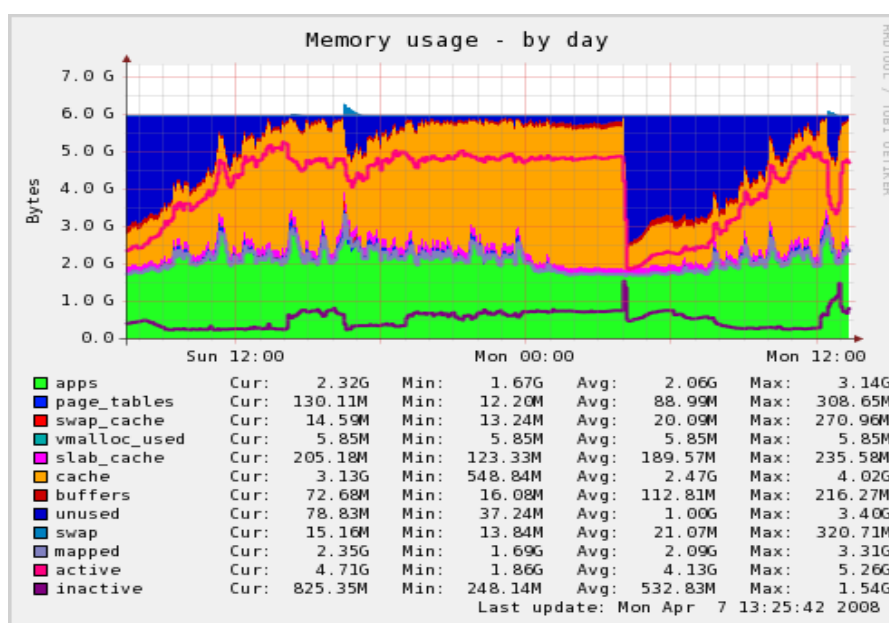


:: [Memory usage](#)

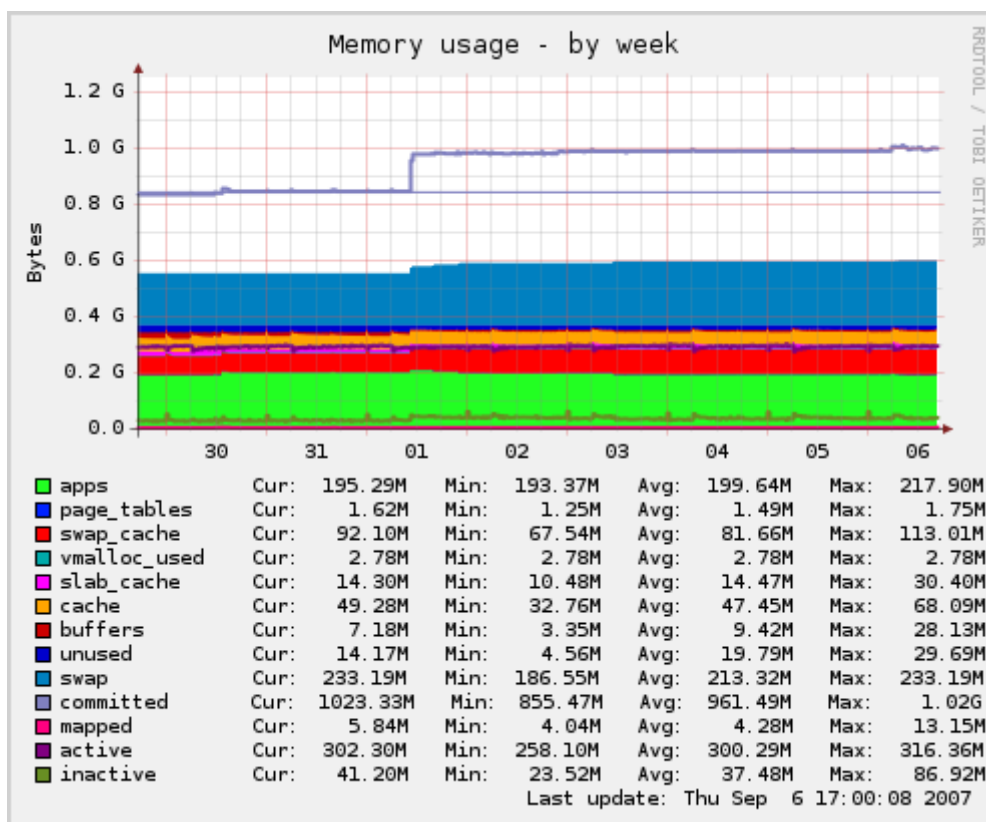


通过 <http://ip/monitoring/> 就可以查看到主机的监控信息

<http://192.168.1.146/monitoring/localhost/localhost.html#Network> 图: Munin 监控信息



图：Munin 监控信息每天



图：Munin 监控信息每周

安装和配置 monit

执行 yum 完成安装

```
yum install monit
```

设置 monit 的启动脚本

```
chkconfig --levels 235 monit on
```

默认的/etc/monit.conf 文件中是注释的示例配置文件，

<http://mmonit.com/wiki/Monit/ConfigurationExamples> 这里是官方提供的配置文件参考。

<http://mmonit.com/monit/documentation/monit.html> 这里是详尽的配置文件的语法格式信息。

大家注意到/etc/monit.conf 文件末尾的信息。

```
#####
#####

## Includes

#####
#####
```

```
##
## It is possible to include additional configuration parts from other files or
## directories.
#
include /etc/monit.d/*
#
#
```

此处的信息的作用是与 apache 的配置文件类似，在/etc/monit.d/中的每个文件都是一个独立的配置文件，便于配置文件的结构化和维护操作。

创建一个简单的配置文件

```
vi /etc/monit.d/monitrc
set daemon 60                #每 2 分钟检查一次
set logfile syslog facility log_daemon    #monit 的日志文件
set mailserver localhost      #设置邮件服务器
set mail-format { from: monit@192.168.1.149 } #出现错误的时候发报警邮件到指定的邮箱
set alert lvsheat@qq.com
set httpd port 2812 and use address 192.168.1.149 #设置 http 监控页面的端口和 ip
    SSL ENABLE
    PEMFILE /var/certs/monit.pem
    allow admin:test          #认证的用户名和密码

#平均负载,内存使用率,cpu 使用率
check system 192.168.1.149
    if loadavg (1min) > 4 then alert
    if loadavg (5min) > 2 then alert
    if memory usage > 75% then alert
    if cpu usage (user) > 70% then alert
    if cpu usage (system) > 30% then alert
    if cpu usage (wait) > 20% then alert
# 磁盘空间使用率
```

check device data with path /dev/sda2

if space usage > 90% then alert

if inode usage > 85% then alert

check device home with path /dev/sda3

if space usage > 85% then alert

if inode usage > 85% then alert

#监控 vsftpd 服务

check process vsftpd with pidfile /var/run/vsftpd.pid

start program = "/etc/init.d/vsftpd start"

stop program = "/etc/init.d/vsftpd stop"

if failed port 21 protocol ftp then restart

if 5 restarts within 5 cycles then timeout

#监控 ssh 服务

check process sshd with pidfile /var/run/sshd.pid

start program "/etc/init.d/sshd start"

stop program "/etc/init.d/sshd stop"

if failed port 22 protocol ssh then restart

if 5 restarts within 5 cycles then timeout

#监控 mysql 服务

check process mysql with pidfile /var/run/mysqld/mysqld.pid

group database

start program = "/etc/init.d/mysqld start"

stop program = "/etc/init.d/mysqld stop"

if failed host 127.0.0.1 port 3306 then restart

if 5 restarts within 5 cycles then timeout

#监控 apache 服务

check process apache with pidfile /var/run/httpd.pid

group www

start program = "/etc/init.d/httpd start"

stop program = "/etc/init.d/httpd stop"

```
if failed host 192.168.1.149 port 80 protocol http
    and request "/monit/testok" then restart
if cpu is greater than 60% for 2 cycles then alert
if cpu > 80% for 5 cycles then restart
if totalmem > 500 MB for 5 cycles then restart
if children > 250 then restart
if loadavg(5min) greater than 10 for 8 cycles then stop
if 3 restarts within 5 cycles then timeout
#监控 postfix 邮件服务
check process postfix with pidfile /var/spool/postfix/pid/master.pid
    group mail
    start program = "/etc/init.d/postfix start"
    stop program = "/etc/init.d/postfix stop"
    if failed port 25 protocol smtp then restart
    if 5 restarts within 5 cycles then timeout
#监控自定义服务
check process web_lb with pidfile /opt/server/web_lb/httpd.pid
    start program = "/opt/bin/lb.sh" #启动脚本
    stop program = "/opt/bin/lb_stop.sh" #停止脚本
    if failed host 192.168.1.149 port 6101 proto http then restart
    if failed host 192.168.1.149 port 6101 proto http for 5 times within 5 cycles then exec
"/opt/bin/lb_pay.sh"
    if failed host 192.168.1.149 port 6102 type TCPSSL proto http then restart
    if failed host 192.168.1.149 port 6102 type TCPSSL proto http for 5 times within 5 cycles
then exec "/opt/bin/lb_pay.sh"
```

这个配置文件监控了 vsftpd, sshd, mysql, apache, postfix 服务, 设定 web 监控的端口是 2812, 设置 lvsheat@qq.com 将接受到警告邮件。

配置 apache 的检查信息

```
mkdir /var/www/monit
echo "hello" > /var/www/monit/token
```


创建/var/certs/monit.pem 信息，设置 https 的 web 监控访问

```
mkdir /var/certs
```

```
cd /var/certs
```

编辑一个默认的 OpenSSL 的配置文件

```
vi /var/certs/monit.cnf
```

```
# create RSA certs - Server
```

```
RANDFILE = ./openssl.rnd
```

```
[ req ]
```

```
default_bits = 1024
```

```
encrypt_key = yes
```

```
distinguished_name = req_dn
```

```
x509_extensions = cert_type
```

```
[ req_dn ]
```

```
countryName = Country Name (2 letter code)
```

```
countryName_default = MO
```

```
stateOrProvinceName      = State or Province Name (full name)
```

```
stateOrProvinceName_default = Monitoria
```

```
localityName              = Locality Name (eg, city)
```

```
localityName_default      = Monittown
```

```
organizationName          = Organization Name (eg, company)
```

```
organizationName_default  = Monit Inc.
```

```
organizationalUnitName     = Organizational Unit Name (eg,  
section)
```

```
organizationalUnitName_default = Dept. of Monitoring  
Technologies
```

```
commonName                 = Common Name (FQDN of your server)
```

commonName_default = server.monit.mo

emailAddress = Email Address

emailAddress_default = root@monit.mo

[cert_type]

nsCertType = server

创建/var/certs/monit.pem 文件

```
openssl req -new -x509 -days 365 -nodes -config ./monit.cnf -out /var/certs/monit.pem  
-keyout /var/certs/monit.pem
```

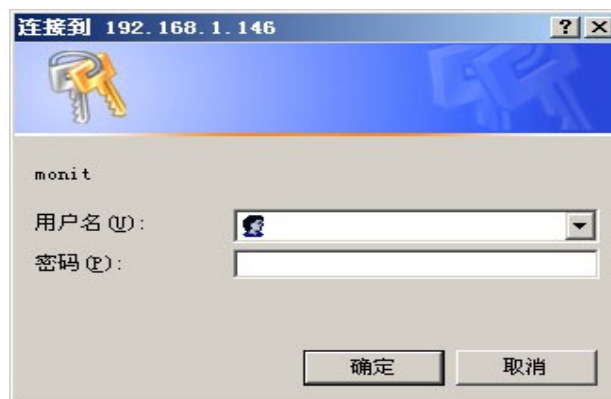
```
openssl gendh 512 >> /var/certs/monit.pem
```

```
openssl x509 -subject -dates -fingerprint -noout -in /var/certs/monit.pem
```

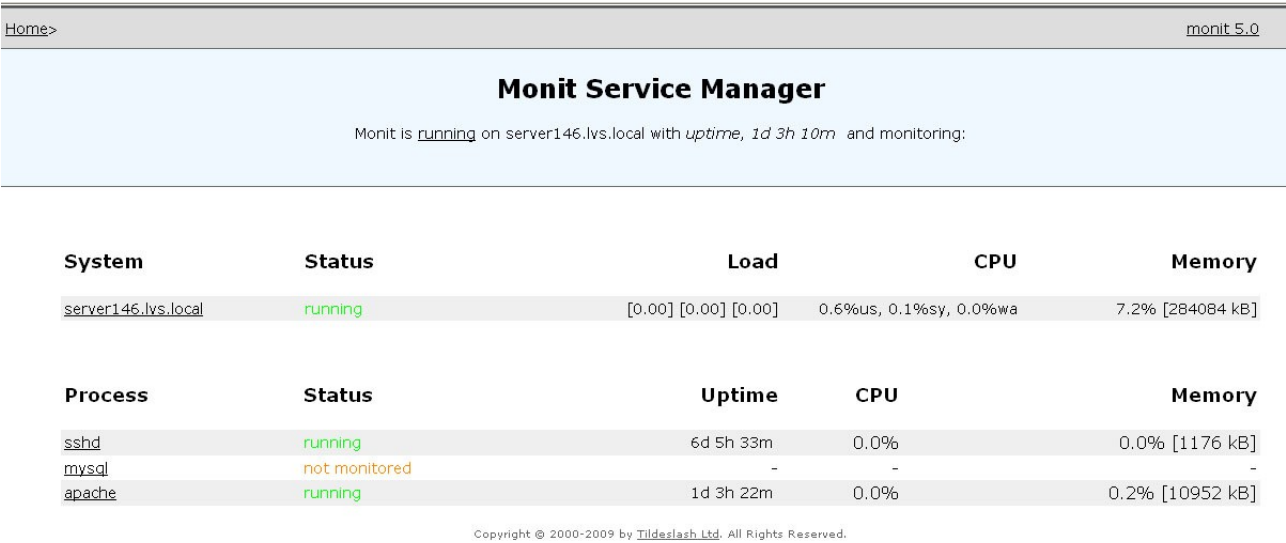
```
chmod 700 /var/certs/monit.pem
```

启动 Monit 服务

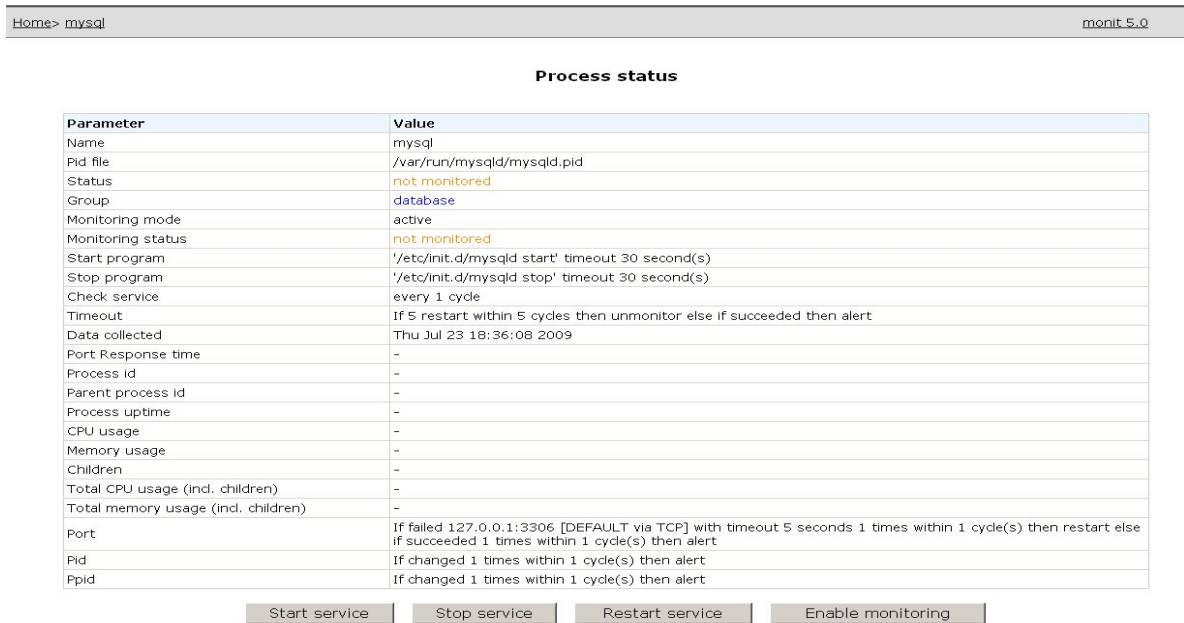
```
/etc/init.d/monit start
```



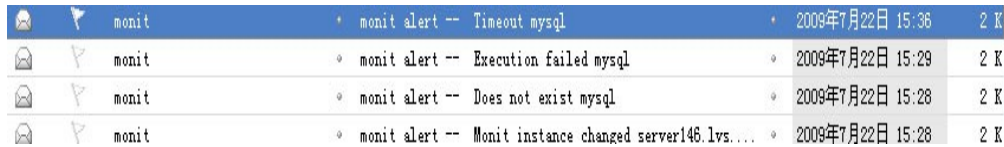
通过 <https://192.168.1.149:2812/> 访问监控页面信息查看邮件可以看到监控的邮件警告信息
图：登陆信息输入 admin 密码 test



图：监控的信息清单



图：监控的服务的详细信息



图：收到的监控警告邮件

```
monit alert -- Timeout mysql
monit
收件人: [REDACTED]
-----
Timeout Service mysql

Date:      Wed, 22 Jul 2009 15:32:47 +0800
Action:    unmonitor
Host:      server146.lvs.local
Description: service timed out and will not be checked anymore

Your faithful employee,
monit
```

图：监控邮件内容

Ntop 的安装和使用

执行 yum 安装 ntop

```
yum install ntop
```

启动 ntop

```
ntop -P /tmp -u nobody
```

ntop startup - waiting for user response!

Please enter the password for the admin user:

Please enter the password again:

第一次执行会要求输入 admin 的 password

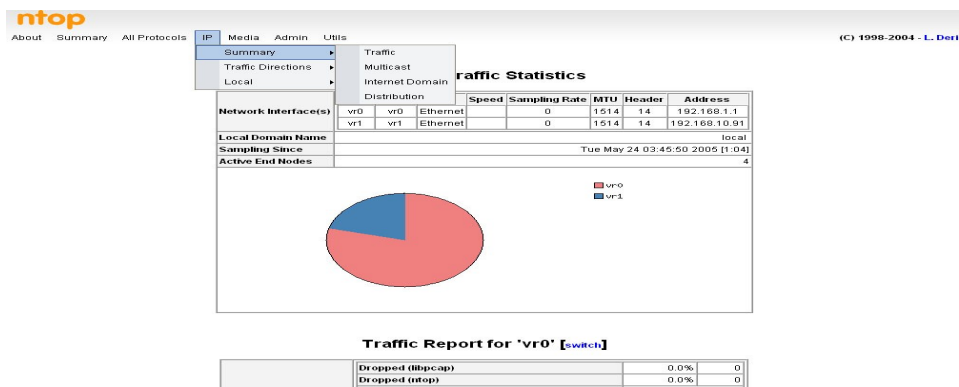
启动后查看本机的端口信息，会出现 TCP3000 的端口

```
netstat -tlnup
```

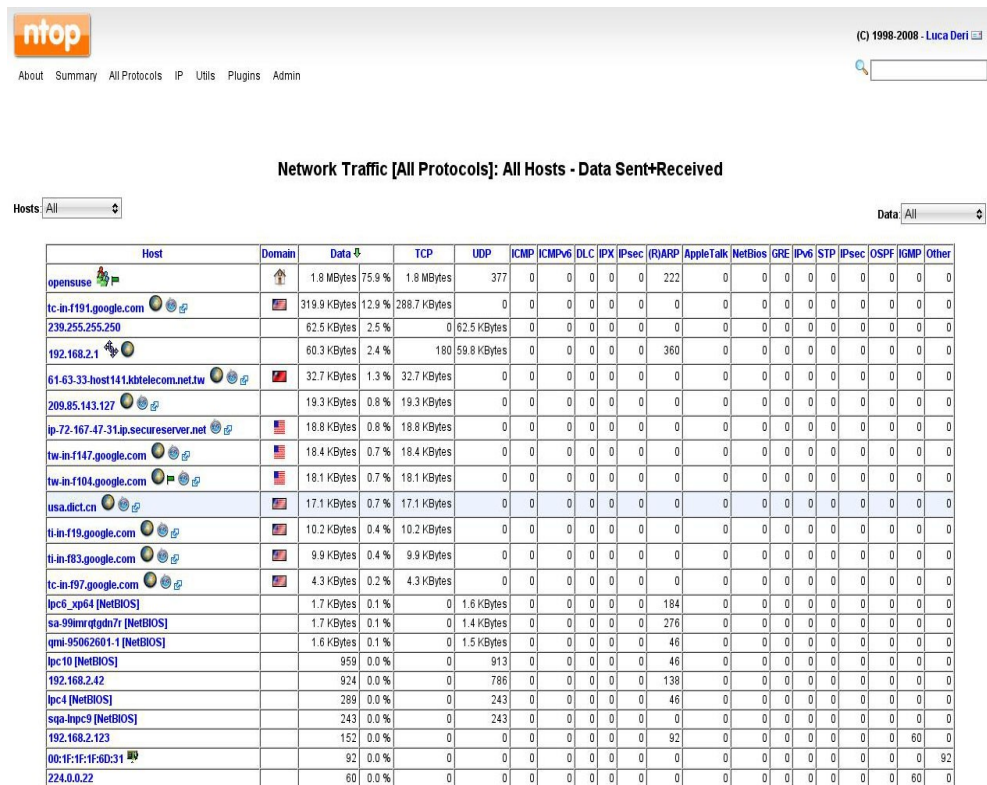
Active Internet connections (only servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program
tcp	0	0	:::3000	:::*	LISTEN	3932/ntop

以在 web 浏览器中输入 `http://ip:3000` 就可以访问到监控的信息页面



图：Ntop 监控信息



图：Ntop 监控信息

现在就可以使用 munin monit ntop 监控 Linux 主机的服务及资源。使用 yum 可以很便捷的的安装软件包，达到快速实现系统服务的目的，维护使用 Centos 的一部分工作就是选择一个很好的 yum 的源，这样可以很快捷的实现管理。

作者简介：CU 网友 kns1024wh，目前从事 Linux 群集方面的具体工作，之前做过多年的 IT 技术支持、MCT 讲师、及 REDFLAG 的技术合作，技术专长群集、unix 主机、AD 部署等，您可以通过电子邮件 lvsheat@qq.com 或者 Chinaunix 社区与他取得联系。

软 Raid 升级—单盘分身大法

ChinaUnix 网友: Jerrywjl

大凡长期混迹于生产环境风口浪尖的哥们，只要一看到 Raid 前面的“软”字估计立马会嗤之以鼻。和硬件 Raid 最大的不同，是各种类型的软件 Raid，甭管是 Host Raid，Dm Raid，板载 Raid 卡等等，说一千道一万都是要依赖于操作系统实现和控制的东西。既然如此，性能和兼容性的问题就成了软 Raid 饱受诟病的先天不足。

的确，在大多数号称关键、要命、企业级和最重要是财大气粗的生产业务环境上，软 Raid 这种东西就如上不得席面的狗肉一样难登大雅之堂（尽管狗肉在当今的确是个好东西……省略感慨和口水数千）。不过话说回来，甭管怎样，这世道还是穷人居多，钱少日子得照过，容灾也一样要考虑，所谓穷价值万贯嘛！因此对于在生产环境中一样要考虑冗余和天灾人祸的小企业和广大善良的第三世界来说，软件 Raid 这种东西自然不乏生存的空间和大把证明自己是个玩意的环境。就像你如果吃不起人参，但至少大可以对着一盘萝卜豪放一把。

至于本人，其实本来并不愿意陷入这种纷争和阶级对立当中。但是无奈受人钱财替人消灾，要是大爷们花了银子，系统上甭管软的硬的有问题和需求都得分析都得看。最后把客人伺候爽了，银子还不是直接落自己口袋！唉！这里我突然想到一个有点不雅的比喻，相信不少老少爷们都已经心照不宣了。呵呵，没办法，世道就这样。借用当今流行的术语，恐怕不少人都是活在这种潜规则中……

刚才说到哪了？Raid，嗯，其实倒霉就在这。提到这个东西个人感觉比较恶心的是：第一，使用软 Raid 的问题居然在所谓的生产环境中层出不穷！这倒是可以理解——毕竟这也是咱天朝和谐社会的主要阶级构成；第二，越来越多的大爷们在潜意识中赋予了软 Raid 无限的小宇宙，大有认为软 Raid 这种东西在生产环境和数据容灾上无往不利，人挡杀人佛挡杀佛之势。这样在无比险恶的生产环境中，对软 Raid 这种玩意的折腾上就多了很多让人无限遐想的玩法。这次要写的血泪史就来源于某位大爷的灵光一现。但也着实让俺长了回见识：

场景很简单，某兄弟有一个生产系统，不过照我估计也就是跑在 PC 机上的生产系统。因为他告诉俺这个生产系统只有一块硬盘，是典型的系统数据和用户数据一锅烩的玩法。

而需求更简单：考虑到来自像七五、十三层、七十码等各种和谐社会的惊喜，准备弄块同样的硬盘，在不重装原来系统和破坏数据的情况下把这个系统从单盘升级到软件 Raid 1。做硬 Raid 没预算，而且也不太好折腾，开不了奔驰夏利总是可以飚一把的吧！？

怎么样？环境和要求够简单的吧？！不过各位听好了，原来的磁盘连个 Raid 毛都沾不上，但现在要从那块毛都没有的单盘彻底脱胎换骨到两块盘的软 Raid 1，而且还必须要保证原来的系统和数据无损，唯一的条件是可以停机操作（呵呵，和没说一样！不停机你来试试！？）所以，应该说这种要求要真弄起来还是有点难度的！没办法，只能抱着试试看的态度，在试验环境中折腾了数把，最终折腾出来的结果也真让俺长了见识！下面就将俺操作的过程全部 show 出来，喜欢玩心跳和找刺激或者晚上总失眠哥们只管找个生产环境照葫芦画瓢验证一把！但丑话先说好，若出了任何问题怨不着俺，要怪就去咱怪天朝社会为什么不提供一个让自己用得硬 Raid 的环境吧！

我的测试环境如下：一台最小化安装的 RHEL4U7 32bit 虚拟机，4G 硬盘。其中/boot 100MB，swap 512MB，/是 2048MB。准备另加一块 4GB 的硬盘和原来的硬盘组成软件 Raid。这就是试验的全部家当，所谓人穷志短，没办法！

首先第一步，将原来硬盘所有的分区类型改成软件 raid，并确保新添加的分区方式完全一摸

一样。下面是经过处理的分区表，不知道怎样将分区类型改成软件 Raid 的哥们就请当我什么都没说，但请务必就此回头是岸！

```
[root@dhcp-0-118 ~]# fdisk -l /dev/sda
```

Disk /dev/sda: 4294 MB, 4294967296 bytes

255 heads, 63 sectors/track, 522 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	fd	Linux raid autodetect
/dev/sda2		14	274	2096482+	fd	Linux raid autodetect
/dev/sda3		275	339	522112+	fd	Linux raid autodetect

接着对于俺这种懒人就用懒办法将/dev/sda 上的分区表来个移形幻影：

```
[root@dhcp-0-118 ~]# sfdisk -d /dev/sda | sfdisk /dev/sdb
```

Checking that no-one is using this disk right now ...

OK

Disk /dev/sdb: 522 cylinders, 255 heads, 63 sectors/track

sfdisk: ERROR: sector 0 does not have an msdos signature

/dev/sdb: unrecognized partition

Old situation:

No partitions found

New situation:

Units = sectors of 512 bytes, counting from 0

Device	Boot	Start	End	#sectors	Id	System
/dev/sdb1	*	63	208844	208782	fd	Linux raid autodetect
/dev/sdb2		208845	4401809	4192965	fd	Linux raid autodetect
/dev/sdb3		4401810	5446034	1044225	fd	Linux raid autodetect
/dev/sdb4		0	-	0	0	Empty

Successfully wrote the new partition table

Re-reading the partition table ...

If you created or changed a DOS partition, /dev/foo7, say, then use dd(1)

to zero the first 512 bytes: dd if=/dev/zero of=/dev/foo7 bs=512 count=1

(See fdisk(8).)

现在两块盘上的分区应该是完全一样了！

下面在原有的三个分区中先对启动分区，即/boot 进行操作。

首先卸载：

```
[root@dhcp-0-118 ~]# umount /boot/
```

然后来一招霸王硬上弓（这里又带来遐想无限……），对/dev/sda1 强行做一个单盘的 Raid：

```
[root@dhcp-0-118 ~]# mdadm -C /dev/md0 -l1 --force -n1 /dev/sda1
```

```
mdadm: /dev/sda1 appears to contain an ext2fs file system
```

```
size=104388K mtime=Fri Jul 24 14:53:25 2009
```

```
Continue creating array? y
```

```
mdadm: array /dev/md0 started.
```

之后将第二块盘的第一个分区加进去升级，让/dev/md0 变成名正言顺的 Raid1：

```
[root@dhcp-0-118 ~]# mdadm /dev/md0 -a /dev/sdb1
```

```
mdadm: hot added /dev/sdb1
```

```
[root@dhcp-0-118 ~]# mdadm -G /dev/md0 -n2
```

```
[root@dhcp-0-118 ~]# cat /proc/mdstat
```

```
Personalities : [raid1]
```

```
md0 : active raid1 sdb1[1] sda1[0]
```

```
104320 blocks [2/2] [UU]
```

```
unused devices: <none>
```

一般来说这步操作应该不会有什么问题，若有问题就请好好反省自己的人品！

确认一下：

```
[root@dhcp-0-118 ~]# df -TH
```

```
Filesystem Type Size Used Avail Use% Mounted on
```

```
/dev/sda2 ext3 2.2G 802M 1.3G 40% /
```

```
none tmpfs 131M 0 131M 0% /dev/shm
```

```
/dev/md0 ext3 104M 13M 86M 13% /boot
```

下面需要重新建立 initrd.img 文件，更新存储这个静态内核模块的镜像文件系统：

```
[root@dhcp-0-118 boot]# mkinitrd -f initrd-2.6.9-78.ELsmp.img $(uname -r)
```

这个操作也一定要完成，否则到待会重启动系统的时候自然会让你叫苦不迭。

现在修改/etc/fstab 和/boot/grub.conf，既然现在是 Raid 了嘛，当然不能再和单盘同日而语了：

下面是原来的文件内容：


```
[root@dhcp-0-118 ~]# cat /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#         all kernel and initrd paths are relative to /boot/, eg.
#         root (hd0,0)
#         kernel /vmlinuz-version ro root=/dev/sda2
#         initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux AS (2.6.9-78.ELsmp)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-78.ELsmp ro root=LABEL=/
    initrd /initrd-2.6.9-78.ELsmp.img
title Red Hat Enterprise Linux AS-up (2.6.9-78.EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-78.EL ro root=LABEL=/
    initrd /initrd-2.6.9-78.EL.img
[root@dhcp-0-118 ~]# cat /etc/fstab
# This file is edited by fstab-sync - see 'man fstab-sync' for details
LABEL=/          /              ext3 defaults    1 1
LABEL=/boot      /boot          ext3 defaults    1 2
none             /dev/pts       devpts gid=5,mode=620 0 0
none             /dev/shm       tmpfs defaults    0 0
none             /proc          proc defaults    0 0
none             /sys           sysfs defaults    0 0
LABEL=SWAP-sda3  swap           swap defaults    0 0
```

```
/dev/hdc          /media/cdrecorder  auto  pamconsole,exec,noauto,managed 0 0
/dev/fd0          /media/floppy      auto  pamconsole,exec,noauto,managed 0 0
```

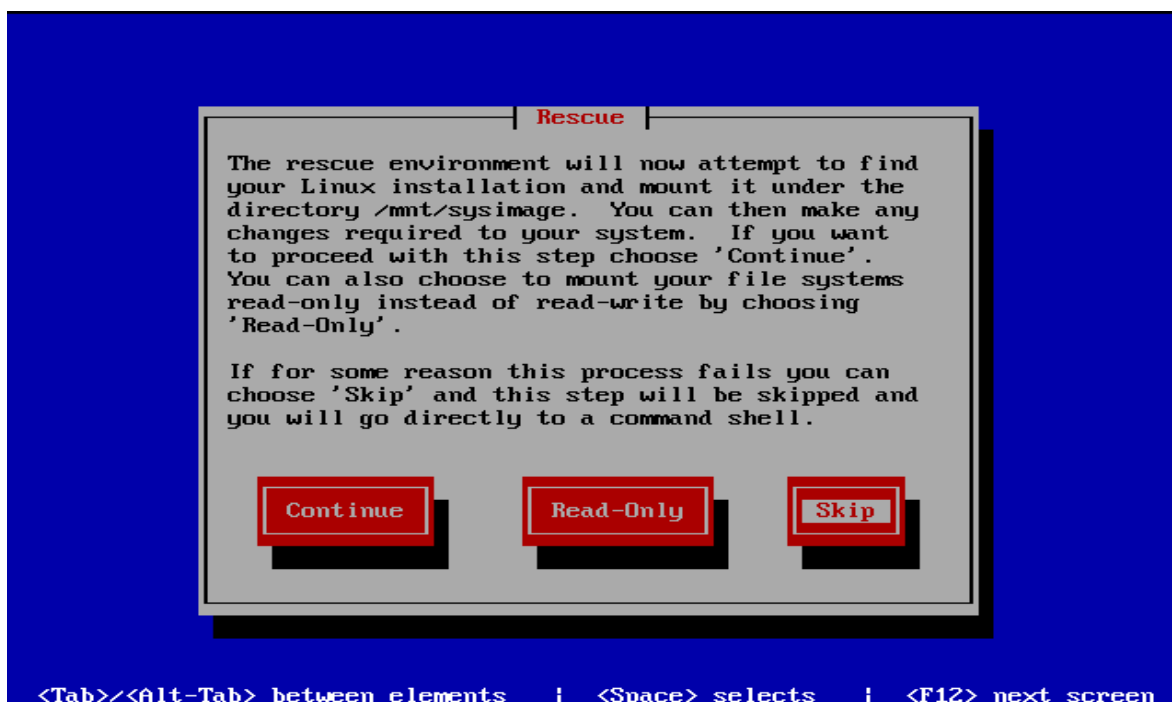
这是修改之后的文件内容：

```
[root@dhcp-0-118 ~]# cat /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#
#   all kernel and initrd paths are relative to /boot/, eg.
#   root (hd0,0)
#   kernel /vmlinuz-version ro root=/dev/sda2
#   initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux AS (2.6.9-78.ELsmp)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-78.ELsmp ro root=/dev/md1
    initrd /initrd-2.6.9-78.ELsmp.img
title Red Hat Enterprise Linux AS-up (2.6.9-78.EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-78.EL ro root=/dev/md1
    initrd /initrd-2.6.9-78.EL.img
[root@dhcp-0-118 ~]# cat /etc/fstab
# This file is edited by fstab-sync - see 'man fstab-sync' for details
/dev/md1          /                  ext3  defaults    1 1
/dev/md0          /boot             ext3  defaults    1 2
none              /dev/pts          devpts gid=5,mode=620 0 0
none              /dev/shm          tmpfs  defaults    0 0
```

```
none          /proc          proc defaults 0 0
none          /sys           sysfs defaults 0 0
# LABEL=SWAP-sda3    swap          swap defaults 0 0
/dev/hdc       /media/cdrecorder  auto pamconsole,exec,noauto,managed 0 0
/dev/fd0       /media/floppy      auto pamconsole,exec,noauto,managed 0 0
```

红色的是俺修改过的。

完成之后，重启系统用光盘引导进入 rescue 模式。针对/dev/sda2 和/dev/sda3 分别建立单个磁盘的软 Raid。在 Rescue 模式的进入过程中选择 skip。



剩下的全部手动操作：

```
# mdadm -C /dev/md1 -l1 --force -n2 /dev/sda2
# mdadm -C /dev/md2 -l1 --force -n2 /dev/sda3
# mdadm /dev/md1 -a /dev/sdb2
# mdadm /dev/md2 -a /dev/sdb3
# mdadm -G /dev/md1 -n2
# mdadm -G /dev/md2 -n2
```

上面的两条命令相信各位看官理解上应该不是什么问题。如果还有很多问题的哥们，恭喜你已经走上了这条不归路，俺只能建议你继续务必理解明白，并及时烧香拜佛祈求佛祖。这里唯一要注意的是，这个时候对于大的文件系统或分区而言，-G是绝对一个漫漫长路，正好有机会思考人生。但不管你产生什么样的冲动，都请务必通过不断查看 cat /proc/mdstat 来确保所有的数据同步都完成再执行下一步操作。否则的话，我将豁出身家性命赌你死得要多难看有多难看！当然在这个

时候很多哥们会开始感悟自己要有个 UPS，那该多好啊.....

在漫长的同步过程彻底完成之后，建议你扫扫机器上的蜘蛛网，并将已经成为 Raid 1 的阵列 mount 起来看看其中的内容是否 OK。如果的确已经 OK，那么恭喜你！你的铤而走险已经成功了一半了。这个 50% 非常关键，因为他预示着你的 Raid 1 阵列已经建立起来。现在则可以重启系统。

不过这个时候纠结的事情来了。通过本人的测试结果发现，其实这个时候你的系统是无法启动起来的。因为在这个时候尽管系统能够正常走过 GRUB，内核模块加载甚至是 init，但在进入到 /etc/fstab 阶段会在挂载根文件系统的时候报告一个很怪异的错误而导致整个系统进入到 repair 模式。在这里特意给大家留下一个截图让各位充分享受一下这种刺激！从报错的信息可以看到，文件系统中的 superblock 报告该文件系统的大小是 524120 个 block，而块设备大小是 524096 个 block。文件系统将块设备塞爆了！？这让俺不禁联想到了伟大天朝尤其是首都的房地产政策和广大第三世界的刚性需求。不过各位大可放心，EXT3 文件系统就算烂透了也比那强百倍，因为至少还能够按需分配，所以我们当然还可以有足够的回旋余地。

```
Setting default font (latarcyrheb-sun16): [ OK ]

Welcome to Red Hat Enterprise Linux AS
Press 'I' to enter interactive startup.
Setting clock (localtime): Fri Jul 24 17:42:15 CST 2009 [ OK ]
Starting udev: [ OK ]
Initializing hardware... storage network audio done [ OK ]
Configuring kernel parameters: [ OK ]
Loading default keymap (us): [ OK ]
Setting hostname localhost.localdomain: [ OK ]
Checking root filesystem
/: The filesystem size (according to the superblock) is 524120 blocks
The physical size of the device is 524096 blocks
Either the superblock or the partition table is likely to be corrupt!

/: UNEXPECTED INCONSISTENCY; RUN fsck MANUALLY.
(i.e., without -a or -p options) [FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue): _
```

不过回到技术方面，其实对于该问题的原因，我只能大概推测，由于软件 Raid 的元数据是要占用一部分的块设备空间，尽管这个占用的空间不大可能也就几十个到几百个块，但这个世界就是充满着不确定性。确定的就像这幅截图所显示的，块设备和文件系统的大小的确还没有聪明到能够自动谈妥条件，因此系统启动就因为这种不一致停在这里过不去了。看来还是得等俺帮他一把。

对于这个问题的解决方法，很多哥们想到了 fsck。呵呵！其实不瞒各位，本人一开始也想到了。但之所以在这个地方纠结是因为即便你 fsck 表面上修复了文件系统，下次重启还是这样。所以修了也是白修。对于这个问题的解决。我们还是要以刚才的方式进入到 rescue 模式中。想必各位已经猜到了，下面就要对文件系统动粗了——请出 resize2fs 缩骨大法！

先在 rescue 模式下对文件系统先进行一次强制修复：

```
# e2fsck -f /dev/sda2
```



```
# e2fsck -f /dev/sdb2
```

然后根据上面报错的提示正式施展传说中的锁骨大法：

```
# resize2fs /dev/sda2 524096
```

```
# resize2fs /dev/sdb2 524096
```

老衲不忘啰嗦一句，这里的 `resize2fs` 的单位是 KB 而不是 block，因为事前俺曾经用 `dumpe2fs` 查看并确认过文件系统块大小是 `block size=1KB`，所以块数量和 KB 数量是一致的。毕竟这个文件系统也就是 2GB 左右。而在很多真实环境中的块大小可能是 4KB 或者是其他值，那么一定要将块数量换算为 KB！所以建议数学不好的哥们就不要铤而走险了，当然如果愿意请再下吃顿饭交流一下感情的话，俺倒是可以豁出去帮你算一下！如果是 MM 的话那俺可能破戒上杆子帮忙算一下。但最后还是那句话：风险自负！善哉善哉！

现在再次重启系统，然后你就会看到.....哇哈哈！还是卡在了 `fstab` 这里！但报错已经不一样了，因为这次是针对 `/dev/sda1` 所在的文件系统，而至于那个命根子根分区，自然已经通过了检查。当然错误类型也还是和刚才一样。但是由于根分区现在已经挂上，且系统启动到这里，`/boot` 文件系统已经完成了他光荣的使命，所以我们就不用再去 `rescue` 模式中纠结了！

```
Initializing hardware... storage network audio done      [ OK ]
Configuring kernel parameters:                          [ OK ]
Loading default keymap (us):                             [ OK ]
Setting hostname localhost.localdomain:                  [ OK ]
Checking root filesystem
/: clean, 30174/262144 files, 205115/524096 blocks      [ OK ]

Remounting root filesystem in read-write mode:           [ OK ]
No RAID disks
Setting up Logical Volume Management:                     [ OK ]
Checking filesystems
/boot: The filesystem size (according to the superblock) is 104388 blocks
The physical size of the device is 104320 blocks
Either the superblock or the partition table is likely to be corrupt!

/boot: UNEXPECTED INCONSISTENCY; RUN fsck MANUALLY.
      (i.e., without -a or -p options)

[FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue): _
```

直接输入管理员密码进入 `repair` 模式，运行命令：

```
# e2fsck -f /dev/sda1
```

```
# e2fsck -f /dev/sdb1
```

并根据刚才的出错提示故伎重演：

```
# resize2fs /dev/sda1 104320
```

```
# resize2fs /dev/sdb1 104320
```

顺便把/etc/fstab 中被禁用掉的 swap 给改回来：

```
# mount -o remount,rw /dev/md1 /
```

```
# vi /etc/fstab
```

将原来注释的 swap 那行改为：

```
/dev/md2 swap swap defaults 0 0
```

保存退出并重建 swap：

```
# mkswap /dev/md2
```

最后别忘了将 grub 在/dev/sdb 上重写一下。为什么要重写 grub？这实际上就是软件 Raid 一个天大的陷阱！因为作为操作系统管理的这种软 Raid 实际上是无法百分百完成设备镜像的。而没有完成同步的部分就在盘头，还恰好就是系统启动的关键，即 MBR 和后续的一部分空间。呵呵，这个世界就是这么奇妙，硬 Raid 虽然贵，但胜在兼容性好并且性能也没人说三道四；软 Raid 尽管经济和高可控性，但类似于这样的命门也足够你恶心一阵子了。所以言归正传，如果没有这部分空间的同步，假如你的第二块盘挂了，那么单靠第一块盘系统自然什么事都没有，这会让很多管理员麻痹大意。不过如果你不好命正好是第一块盘中招，那系统会连 GRUB 都进不去！这种恐怖的场景足以让很多菜鸟管理员一身冷汗了。所以既然如此，就将一切可能发生的情况预料在前。

```
GNU GRUB version 0.95 (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported. For the first word, TAB
  lists possible command completions. Anywhere else TAB lists the possible
  completions of a device/filename.]

grub> root (hd1,0)
Filesystem type is ext2fs, partition type 0xfd

grub> setup (hd1)
Checking if "/boot/grub/stage1" exists... no
Checking if "/grub/stage1" exists... yes
Checking if "/grub/stage2" exists... yes
Checking if "/grub/e2fs_stage1_5" exists... yes
Running "embed /grub/e2fs_stage1_5 (hd1)"... 16 sectors are embedded.
succeeded
Running "install /grub/stage1 (hd1) (hd1)1+16 p (hd1,0)/grub/stage2 /grub/grub
.conf"... succeeded
Done.

grub> quit_
```

完成这步操作之后，终于可以放心重启整个系统。重启过程正常情况下不会有任何问题，完成之后显示阵列信息，一切 OK！这次的分身大法终于彻底完成！

```
[root@dhcp-0-118 ~]# cat /proc/mdstat
```

```
Personalities : [raid1]
```

```
md1 : active raid1 sdb2[1] sda2[0]
      2096384 blocks [2/2] [UU]
md2 : active raid1 sdb3[1] sda3[0]
      522048 blocks [2/2] [UU]
md0 : active raid1 sdb1[1] sda1[0]
      104320 blocks [2/2] [UU]
unused devices: <none>
[root@dhcp-0-118 ~]# df -TH
Filesystem  Type  Size  Used Avail Use% Mounted on
/dev/md1    ext3   2.2G  805M  1.3G  41% /
/dev/md0    ext3   104M   13M   87M  13% /boot
none        tmpfs  131M    0  131M   0% /dev/shm
[root@dhcp-0-118 ~]# mdadm -D --scan > /etc/mdadm.conf
[root@dhcp-0-118 ~]# cat /etc/mdadm.conf
ARRAY /dev/md1 level=raid1 num-devices=2 UUID=f4e187fb:8a2ae9f5:689c862d:197ed8e5
ARRAY /dev/md2 level=raid1 num-devices=2 UUID=684e1a26:a570b86e:ffc29636:3d25bf08
ARRAY /dev/md0 level=raid1 num-devices=2
UUID=6a743391:49212382:fd830cd8:316b7ba5
```

纵观整个操作的过程可以发现，尽管需求很简单，但是在实际操作过程中会出现各种类型的意外，对于意外的观察和处理，务必要做到胆大心细。而且如果哪位哥们真敢在你的生产系统上赴汤蹈火的话，还需要比我考虑更多的因素，如 Raid 同步的数据量，停机时间，原有磁盘是否还有 LVM，是否在整个环境中还有 UPS 这样的护身符等等。反正说一千道一万，我还是那句话：真的希望各位在做方案或者部署的时候凡事想在前头，我这么折腾也是被人逼出来。否则这种顾头不顾腩的事情还是建议各位少干。

带你剖析 Linux 目录结构

随着 Linux 的不断发展，越来越多的人开始使用 Linux，对于那些刚刚接触的人来说，恐怕最先感到困惑的就是那些“不明不白”的目录了。如果想熟练使用 Linux，让 Linux 听命于自己，就必须掌握这些目录，下面就以 XteamLinux 为例，介绍一下在该系统下的目录。

/bin bin 是 Binary 的缩写。这个目录存放着最经常使用的命令。

/boot 这里存放的是启动 Linux 时使用的一些核心文件，包括一些链接文件以及镜像文件。

/dev dev 是 Device(设备)的缩写。该目录下存放的是 Linux 的外部设备，在 Linux 中访问设备的方式和访问文件的方式是相同的。

/etc 这个目录用来存放所有的系统管理所需要的配置文件和子目录。

/home 用户的主目录，在 Linux 中，每个用户都有一个自己的目录，一般该目录名是以用户的账号命名的。

/lib 这个目录里存放着系统最基本的动态链接共享库，其作用类似于 Windows 里的 DLL 文件。几乎所有的应用程序都需要用到这些共享库。

/lost+found 这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些文件。

/mnt 在这里面中有四个目录，系统提供这些目录是为了让用户临时挂载别的文件系统的，我们可以将光驱挂载在 /mnt/cdrom 上，然后进入该目录就可以查看光驱里的内容了。

/proc 这个目录是一个虚拟的目录，它是系统内存的映射，我们可以通过直接访问这个目录来获取系统信息。这个目录的内容不在硬盘上而是在内存里，我们也可以直接修改里面的某些文件，比如可以通过下面的命令来屏蔽主机的 ping 命令，使别人无法 ping 你的机器：

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_
ignore_all。
```

/root 该目录为系统管理员，也称作超级权限者的用户主目录。

/sbin s 就是 Super User 的意思，这里存放的是系统管理员使用的系统管理程序。

/tmp 这个目录是用来存放一些临时文件的。

我们要用到的很多应用程序和文件几乎都存放在 usr 目录下。具体来说：

/usr/X11R6 存放 X-Windows 的目录；

/usr/games 存放着 XteamLinux 自带的小游戏；

/usr/bin 存放着许多应用程序；

/usr/sbin 存放 root 超级用户使用的管理程序；

/usr/doc Linux 技术文档；

/usr/include 用来存放 Linux 下开发和编译应用程序所需要的头文件；

/usr/lib 存放一些常用的动态链接共享库和静态档案库；

/usr/local 这是提供给一般用户的 /usr 目录，在这里安装一般的应用软件；

/usr/man 帮助文档所在的目录；

/usr/src Linux 开放的源代码，就存在这个目录，爱好者们别放过哦；

/var 这个目录中存放着在不断扩充着的东西，我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。如果你想做一个网站，你也会用到 /var/www 这个目录。

本文简单讲解了一下目录的大致内容，如果想成为 Linux 高手，则还要进一步学习 Linux 下的这些目录。

不输入密码自动通过 SSH 方式登录服务器

ChinaUnix 网友: Seaquester

看到 ChinaUnix 上面有一个网友的帖子，想在 linux 下使用 ssh 登录的时候不用每次都输入密码，又不能使用 key 的方式。在网上搜索了一下，有网友用 expect 写了一个自动登录的脚本，但是我试过之后，却发现不能用。

后来看到有人说使用 sshpass 可以解决问题。所以自己写了一个脚本。脚本写的比较简单，没有考虑安全性的问题。要使用这个脚本必须安装 sshpass 这个软件。

sshlogin.sh:

```
#!/bin/bash
#=====
#
# FILENAME:
#   sshlogin.sh
#
# DESCRIPTION:
#   Script to use user and password from config file to login remote
#   ssh server.
#   This script needs 1 argument to login the remote ssh server:
#   ipaddr = IP Address of remote ssh server.
#
#   For example:
#   ./sshlogin.sh 192.168.0.1
#
#   You need to create a config file(pass.dat) with entries like:
#   server1 | user | password
#   server2 | user | password
#
# REVISION(MM/DD/YYYY):
#   07/01/2009 Shengkui Leng (shengkui.leng@advantech.com.cn)
#   - Initial version
#
#=====

#-----
# NAME:
#   print_usage
#
```

```
# DESCRIPTION:
#   Print usage information
#
# PARAMETERS:
#   $1 - Program name
#
# RETURNS:
#   None
#-----
print_usage ()
{
    echo "Usage: $1 <Server IP>"
    exit 1
}
[ $# -eq 1 ] || print_usage $0
```

```
CONFIG_FILE=pass.dat
SERVER=$1
ipaddr=""
username=""
password=""
```

```
found=0
```

```
while read line ; do
    ipaddr=$(echo $line | cut -d' ' -f1)
    username=$(echo $line | cut -d' ' -f2)
    password=$(echo $line | cut -d' ' -f3)
```

```
    if [ "$SERVER" == "$ipaddr" ] ; then
        #The server found!
        found=1
        break
    fi
```

```
done < $CONFIG_FILE
```

```
if [ $found -eq 0 ] ; then
    echo "The server not found!"
    exit 2
fi
```

```
if [ -z "$password" ] || [ -z "$ipaddr" ] || [ -z "$username" ] ; then
    echo "Invalid config file: $CONFIG_FILE!"
    exit 3
fi

#Automatically add new host keys to user known hosts files(known_hosts)
AUTO_ADD_HOST_KEY="-oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no"

echo "Logining(ssh) $username@$ipaddr..."
sshpass -p "$password" ssh $username@$ipaddr $AUTO_ADD_HOST_KEY

exit 0
```

注 1：需要建立一个记录密码、用户名和 ssh server IP 的文件 pass.dat, 格式为：IP | UserName | Password。示例如下：

```
pass.dat:
192.168.0.1 | root | a123B56
192.168.0.2 | root | 7X890Yz
```

注 2：如果是第一次用 ssh 登录某一个 server 时，ssh 会提示：

```
The authenticity of host '192.168.0.1 (192.168.0.11)' can't be established.
RSA key fingerprint is 7b:b9:c3:cd:01:cd:2f:19:4e:96:d3:66:27:55:7f:65.
Are you sure you want to continue connecting (yes/no)?
```

所以上面的脚本在调用 ssh 时指定了参数：

```
-oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no
```

这样 ssh 就不会出现上面的提示，详情请参考 man ssh_config。

RHEL5 实现笔记本电脑触摸板开关功能

ChinaUnix 网友：carrylife

本人有一台笔记本是 DELL D630，今天不是讨论它的性能怎样的。从我买它的那一天起，里面装的就是 linux 操作系统，有装过 suse desktop 版，也装过 ubuntu 的，但适合它身份的莫过于 RHEL 版，两者的商务风格不谋而合，很是般配。

不过，问题也来了，这台本本在 windows 下用时，有 DELL 的专属程序控制 tochpad 的启用或关闭，但在 RHEL5 下面，让我着实受不了，没有什么 专有程序控制它，电脑本身也没有提供硬件开关，所以在操作时，不小时碰到触摸板时，mouse 指针就乱飞，有时会将打的文字顺序全乱，想想总会有办法来解决它的。

linux 下的触摸板驱动是由 syndaemon 来支持的，所以就拿它来做手脚了...

首先我们需要了解一些东西，就是 syndaemon 是通过由驱动程序建立的共享内存区域来访问

synaptics 触摸设备，这个非常重要，默认情况下 syndaemon 和共享内存区 的通讯是关闭的。所以必须先要将它们的通讯打开，否则在启动 syndaemon 时会报错而无法继续实现我们的目的。如果要建立 syndaemon 和共享 内存区的通讯，需要修改文件/etc/X11/xorg.conf 文件，找到"InputDevice"部分并按照下列格式增加"SHMCconfig" 选项

```
Section "InputDevice"
    Identifier   "Synaptics"
    Driver       "synaptics"
    Option       "Device" "/dev/input/mice"
    Option       "Protocol" "auto-dev"
    Option       "Emulate3Buttons" "yes"
    Option       "SHMConfig" "on"      - - - - 开启共享内存区的通讯
    Option       "LeftEdge" "120"
    Option       "RightEdge" "830"
    Option       "TopEdge" "120"
    Option       "BottomEdge" "650"
```

EndSection

然后保存退出。

注销并重新登录系统后，共享内存区的通讯就开启了。

然后接下来，我们就可以来写实现 touchpad 开关的部分了

在用户配置文件中可以设定，默认进入系统后，关闭触摸板,为了达到这个目的，我们可以在用户配置文件中做如下指定，步骤如下：

```
$vim ~/.bashrc

# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific aliases and functions

synclient touchpadoff=1 -----禁用触摸板
```

然后，保存并退出编辑器

使用现有的账户，重新登录系统即可，默认的触摸板就是关闭的了。

这样子，如果我们没有外接 MOUSE 时又不方便了，难道还要将上面新加的内容去掉，再重新进行 login 吗？当然不用，我们利用上面得到的知识，写一个小程 序放在~/.bashrc 下，以达到程序控制触摸板软开关的目的，这样子，可以自由切换，当然，可以在用户全局配置文件 /etc/bashrc 中写，这样可以避免不同用户进入系统时情形不一的情况。

```
$vim ~/.bashrc
```


以下是我打开并编辑后的内容

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific aliases and functions

function touchpad()
{
    if [ "$1" == "on" ]; then
        synclient touchpadoff=0
    elif [ "$1" == "off" ]; then
        synclient touchpadoff=1
    fi
}
```

然后完成后保存退出，重新进入系统或是重新启动 X 服务即可实现前面预想的功能了。
当再次进入系统后，我们使用如下命令打开或关闭 tochpad:

```
$touchpad on
$touchpad off
```

最后，说明一下，我的本本除了触摸板外，还有指点设备，和 Tinkpad 的类似，不过 DELL 的商务机是“小蓝帽”，在我实现上述功能时，并没有能成功关闭 指点设备的使用，这是由于它们使用不同的模块驱动所致。不过，指点设备不用关，想必用了 linux+指点设备的朋友们，都能感觉到，它们是绝配！

如果你也有类似情况，试一下，享受 linux 给你带来的乐趣吧！

Linux 下 patch 的制作和应用

首先介绍一下 diff 和 patch。在这里不会把 man 在线文档上所有的选项都介绍一下，那样也没有必要。在 99%的时间里，我们只会用到几个选项。所以必须学会这几个选项。

1、diff

NAME

diff - find differences between two files

SYNOPSIS

diff [options] from-file to-file

简单的说，diff 的功能就是用来比较两个文件的不同，然后记录下来，也就是所谓的 diff 补丁。语法格式：diff 【选项】 源文件（夹） 目的文件（夹），就是要给源文件（夹）打个补丁，使之变成目的文件（夹），术语也就是“升级”。下面介绍三个最为常用选项：

-r 是一个递归选项，设置了这个选项，diff 会将两个不同版本源代码目录中的所有对应文件全部都进行一次比较，包括子目录文件。

-N 选项确保补丁文件将正确地处理已经创建或删除文件的情况。

-u 选项以统一格式创建补丁文件，这种格式比缺省格式更紧凑些。

2、patch

NAME

patch - apply a diff file to an original

SYNOPSIS

patch [options] [originalfile [patchfile]]

but usually just

patch -pnum <patchfile>

简单的说，patch 就是利用 diff 制作的补丁来实现源文件（夹）和目的文件（夹）的转换。这样说就意味着你可以有源文件（夹）——>目的文件（夹），也可以目的文件（夹）——>源文件（夹）。下面介绍几个最常用选项：

-p0 选项要从当前目录查找目的文件（夹）

-p1 选项要忽略掉第一层目录，从当前目录开始查找。

在这里以实例说明：

--- old/modules/pcitable Mon Sep 27 11:03:56 1999

+++ new/modules/pcitable Tue Dec 19 20:05:41 2000

如果使用参数-p0，那就表示从当前目录找一个叫做 old 的文件夹，在它下面寻找 modules 下的 pcitable 文件来执行 patch 操作。

如果使用参数-p1，那就表示忽略第一层目录（即不管 old），从当前目录寻找 modules 的文件夹，在它下面找 pcitable。这样的前提是当前目录必须为 modules 所在的目录。而 diff 补丁文件则可以在任意位置，只要指明了 diff 补丁文件的路径就可以了。当然，可以用相对路径，也可以用绝对路径。不过我一般习惯用相对路径。

-E 选项说明如果发现了空文件，那么就删除它

-R 选项说明在补丁文件中的“新”文件和“旧”文件现在要调换过来了（实际上就是给新版本打补丁，让它变成老版本）

下面结合具体实例来分析和解决，分为两种类型：为单个文件打补丁和为文件夹内的多个文件打补丁。

环境：在 RedHat 9.0 下面以 armlinux 用户登陆。

目录树如下：

```
|-- bootloader
|-- debug
|-- images
|-- kernel
|-- program
|-- rootfiles
|-- software
|-- source
|-- sysapps
|-- tmp
`-- tools
```

下面在 program 文件夹下面建立 patch 文件夹作为实验用，然后进入 patch 文件夹。

一、为单个文件进行补丁操作

1、建立测试文件 test0、test1

```
[armlinux@lqm patch]$ cat >>test0<<EOF
```

```
> 111111
```

```
> 111111
```

```
> 111111
```

```
> EOF
```

```
[armlinux@lqm patch]$ more test0
```

```
111111
```

```
111111
```

```
111111
```

```
[armlinux@lqm patch]$ cat >>test1<<EOF
```

```
> 222222
```

```
> 111111
```

```
> 222222
```

```
> 111111
```

```
> EOF
```

```
[armlinux@lqm patch]$ more test1
```

```
222222
```

```
111111
```

```
222222
```

```
111111
```

2、使用 diff 创建补丁 test1.patch

```
[armlinux@lqm patch]$ diff -uN test0 test1 > test1.patch
```

【注：因为单个文件，所以不需要-r 选项。选项顺序没有关系，即可以是-uN，也可以是-NU。】

```
[armlinux@lqm patch]$ ls
```

```
test0 test1 test1.patch
```

```
[armlinux@lqm patch]$ more test1.patch
```

```
*****
```

patch 文件的结构

补丁头

补丁头是分别由---/+++开头的两行，用来表示要打补丁的文件。---开头表示旧文件，+++开头表示新文件。

一个补丁文件中的多个补丁

一个补丁文件中可能包含以---/+++开头的很多节，每一节用来打一个补丁。所以在一个补丁文件中可以包含好多个补丁。

块

块是补丁中要修改的地方。它通常由一部分不用修改的东西开始和结束。他们只是用来表示要修改的位置。他们通常以@@开始，结束于另一个块的开始或者一个新的补丁头。

块的缩进

块会缩进一列，而这一列是用来表示这一行是要增加还是要删除的。

块的第一列

+号表示这一行是要加上的。

-号表示这一行是要删除的。

没有加号也没有减号表示这里只是引用的而不需要修改。

```
*****
```

diff 命令会在补丁文件中记录这两个文件的首次创建时间，如下

```
--- test0    2006-08-18 09:12:01.0000000000 +0800
```

```
+++ test1    2006-08-18 09:13:09.0000000000 +0800
```

```
@@ -1,3 +1,4 @@
```

```
+222222
```

```
111111
```

```
-111111
```

```
+222222
```

```
111111
```

```
[armlinux@lqm patch]$ patch -p0 < test1.patch
```

```
patching file test0
```

```
[armlinux@lqm patch]$ ls
```

```
test0 test1 test1.patch
```

```
[armlinux@lqm patch]$ cat test0
```

```
222222
```

```
111111
```

```
222222
```

```
111111
```

3、可以去除补丁，恢复旧版本

```
[armlinux@lqm patch]$ patch -RE -p0 < test1.patch
```

```
patching file test0
```

```
[armlinux@lqm patch]$ ls
```

```
test0 test1 test1.patch
```

```
[armlinux@lqm patch]$ cat test0
```

```
111111
```

```
111111
```

```
111111
```

二、为多个文件进行补丁操作

1、创建测试文件夹

```
[armlinux@lqm patch]$ mkdir prj0
```

```
[armlinux@lqm patch]$ cp test0 prj0
```



```
[armlinux@lqm patch]$ ls
prj0 test0 test1 test1.patch
[armlinux@lqm patch]$ cd prj0/
[armlinux@lqm prj0]$ ls
test0
[armlinux@lqm prj0]$ cat >>prj0name<<EOF
> -----
> prj0/prj0name
> -----
> EOF
[armlinux@lqm prj0]$ ls
prj0name test0
[armlinux@lqm prj0]$ cat prj0name
-----
prj0/prj0name
-----
[armlinux@lqm prj0]$ cd ..
[armlinux@lqm patch]$ mkdir prj1
[armlinux@lqm patch]$ cp test1 prj1
[armlinux@lqm patch]$ cd prj1
[armlinux@lqm prj1]$ cat >>prj1name<<EOF
> -----
> prj1/prj1name
> -----
> EOF
[armlinux@lqm prj1]$ cat prj1name
-----
prj1/prj1name
-----
[armlinux@lqm prj1]$ cd ..
```

2、创建补丁

```
[armlinux@lqm patch]$ diff -uNr prj0 prj1 > prj1.patch
[armlinux@lqm patch]$ more prj1.patch
diff -uNr prj0/prj0name prj1/prj0name
--- prj0/prj0name    2006-08-18 09:25:11.000000000 +0800
+++ prj1/prj0name    1970-01-01 08:00:00.000000000 +0800
@@ -1,3 +0,0 @@
-----
-prj0/prj0name
-----
diff -uNr prj0/prj1name prj1/prj1name
--- prj0/prj1name    1970-01-01 08:00:00.000000000 +0800
+++ prj1/prj1name    2006-08-18 09:26:36.000000000 +0800
@@ -0,0 +1,3 @@
+-----
+prj1/prj1name
+-----
diff -uNr prj0/test0 prj1/test0
--- prj0/test0 2006-08-18 09:23:53.000000000 +0800
+++ prj1/test0 1970-01-01 08:00:00.000000000 +0800
@@ -1,3 +0,0 @@
-111111
-111111
-111111
diff -uNr prj0/test1 prj1/test1
--- prj0/test1 1970-01-01 08:00:00.000000000 +0800
+++ prj1/test1 2006-08-18 09:26:00.000000000 +0800
@@ -0,0 +1,4 @@
+222222
+111111
+222222
+111111
```

```
[armlinux@lqm patch]$ ls
prj0 prj1 prj1.patch test0 test1 test1.patch
[armlinux@lqm patch]$ cp prj1.patch ./prj0
[armlinux@lqm patch]$ cd prj0
[armlinux@lqm prj0]$ patch -p1 < prj1.patch
patching file prj0name
patching file prj1name
patching file test0
patching file test1
[armlinux@lqm prj0]$ ls
prj1name prj1.patch test1
[armlinux@lqm prj0]$ patch -R -p1 < prj1.patch
patching file prj0name
patching file prj1name
patching file test0
patching file test1
[armlinux@lqm prj0]$ ls
prj0name prj1.patch test0
```

总结一下:

单个文件

```
diff -uN from-file to-file >to-file.patch
```

```
patch -p0 < to-file.patch
```

```
patch -RE -p0 < to-file.patch
```

多个文件

```
diff -uNr from-docu to-docu >to-docu.patch
```

```
patch -p1 < to-docu.patch
```

```
patch -R -p1 <to-docu.patch
```

使用 qemu 进行内核源码级调试

ChinaUnix 网友: Chinahucai

内核源码调试对于内核初学者是一件有一定难度的事。工欲善其事，必先利其器，要想成功地进行内核源码级的调试，首先，必须找到一个合适的工具，下面，笔者就来介绍内核源码调试的一款工具 QEMU。

QEMU 是一个通用并开放源代码的模拟器，其功能相当的强大，例如：可以用 QEMU 来模拟一个完整的系统，同时，也可以用 QEMU 来实现系统源码级的调试。如果您想对 QEMU 仿真器有更加深入的了解，请参阅其官方网站：

<http://www.qemu.org/>

下面笔者从如何得到 QEMU，以及如何在 linux 下安装 QEMU 并进行源码级的调试做一个详细的介绍。

(一) qemu 的获得以及安装

得到 qemu 是相当方便的，到其官方网站

<http://www.nongnu.org/qemu/download.html> 下载 QEMU Linux

下载最新版本即可。

接下来在 linux 安装 qemu 的详细步骤：

第一步：

把下载的文件放到工作目录下，解压缩：

例如：

```
huanghucai@huanghucai-laptop:~/kernel_learning$ tar zxvf qemu-0.10.5.tar.gz
```

第二步：切换到 qemu 目录下

```
huanghucai@huanghucai-laptop:~/kernel_learning$ cd qemu-0.10.5
```

第三步：配置安装

```
huanghucai@huanghucai-laptop:~/kernel_learning/qemu-0.10.5$ ./configure
```

```
huanghucai@huanghucai-laptop:~/kernel_learning/qemu-0.10.5$ sudo make
```

```
huanghucai@huanghucai-laptop:~/kernel_learning/qemu-0.10.5$ sudo make install
```

这样，qemu 就安装好了，安装好了之后，对于一个新的软件，我们需要查看其具体的使用方法，请查看它的 man 手册。

(二) 使用 QEMU 进行内核源码级调试

接下来，就是利用 QEMU 进行内核源码调试。要想在 QEMU 下进行内核源码调试，和其他内核调试方式一样，首先是准备内核镜像。为了能够进行源码级的跟踪、调试，需要一个包含调试信息的内核镜像。

下面将会以 linux-2.6.23.2 内核版本为例来进行讲解。

首先是到 linux 内核官方网站 www.kernel.org 获得 linux-2.6.23.2 内核源码 [linux-2.6.23.2.tar.gz](#)，下载后解压缩，将源代码放到工作目录下，

例如：放到目录 `huanghucai@huanghucai-laptop:~/kernel_learning$`

(这里说点题外话，这里笔者推荐一款下载工具 axel，下载速度很快的，为了尽快的得到源代码，安装 axel，复制链接，之后终端执行下面命令：

axel -n 50 -o . http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.23.2.tar.gz 不过 60 秒，你就可以得到 linux 内核源码了，上面的 . 表示你把下载源代码放在了当前工作目录下，使用了 50 线程，要想对 axel 有深入的了解，你可以 man 一下就 Ok 了)

下载内核源代码之后，解压缩，并进入内核源码目录：

```
huanghucai@huanghucai-laptop:~/kernel_learning$ tar zxvf linux-2.6.23.2
linux-2.6.23.2.tar.gz
huanghucai@huanghucai-laptop:~/kernel_learning$ cd linux-2.6.23.2
```

第一步、生成新的.config 文件，其中的配置采用默认选项就可以了

```
huanghucai@huanghucai-laptop:~/kernel_learning/linux-2.6.23.2$ defconfig
```

第二步、修改已有的.config 文件，在这个步骤中，添加新的内核选项

```
huanghucai@huanghucai-laptop:~/kernel_learning/linux-2.6.23.2$ make menuconfig
```

进入顶级选项 Kernel hacking 选项，选中 kernel debugging, 之后再次选中 Compile kernel with debug info 选项，选择以后出来就可以了；

第三步、创建大内核镜像 bzImage

```
huanghucai@huanghucai-laptop:~/kernel_learning/linux-2.6.23.2$ make bzImage
```

第三步运行结束需要较长的时间，当第三步完成之后，如果没有出现什么问题，就可以直接略去下面的这些内容跳到第四步就 OK 了；如果你使用的是 gcc 4.3，那么很有可能会出现创建失败的情况，错误提示如下所示：

```
.....
.....
LD    .tmp_vmlinux1
kernel/built-in.o: In function `getnstimeofday':
(.text+0x1b2b1): undefined reference to `__umoddi3'
kernel/built-in.o: In function `do_gettimeofday':
(.text+0x1b36c): undefined reference to `__udivdi3'
kernel/built-in.o: In function `do_gettimeofday':
(.text+0x1b38f): undefined reference to `__umoddi3'
kernel/built-in.o: In function `timekeeping_resume':
timekeeping.c:(.text+0x1b520): undefined reference to `__udivdi3'
timekeeping.c:(.text+0x1b543): undefined reference to `__umoddi3'
kernel/built-in.o: In function `update_wall_time':
(.text+0x1bb9d): undefined reference to `__udivdi3'
kernel/built-in.o: In function `update_wall_time':
(.text+0x1bbc0): undefined reference to `__umoddi3'
kernel/built-in.o: In function `update_wall_time':
(.text+0x1bc57): undefined reference to `__udivdi3'
kernel/built-in.o: In function `update_wall_time':
(.text+0x1bc81): undefined reference to `__umoddi3'
make: *** [.tmp_vmlinux1] Error 1
```

这个错误的原因是因为 gcc 4.3 在处理 64 位整数运算的时候出现的问题，解决的办法据我所知有两

种，这里介绍其中一种。在 linux2.6.23.2 目录下的 Makefile 文件中给变量 CFLAGS_KERNEL 赋值为：-fno-tree-scev-cprop，

改变值之后重新编译一次就 OK 了。如果出现了 undefined reference to `__stack_chk_fail'，可以在文件 Makefile 中的 CFLAGS 设置值

-fno-stack-protector，即关闭掉栈的保护。

如果出现其他编译失败的情况，你可以借助于 google 找到相关的解决方案。

第四步就是准备根文件系统镜像，这里我从 qemu 的官方网站下载了 linux-0.2.img.bz2，解压缩并将解压缩后的文件放到源代码目录下。

经过上面四个步骤的工作，现在你已经将内核源码级的调试环境搭建完了，下一步就来进入 qemu 下的内核源码级调试。

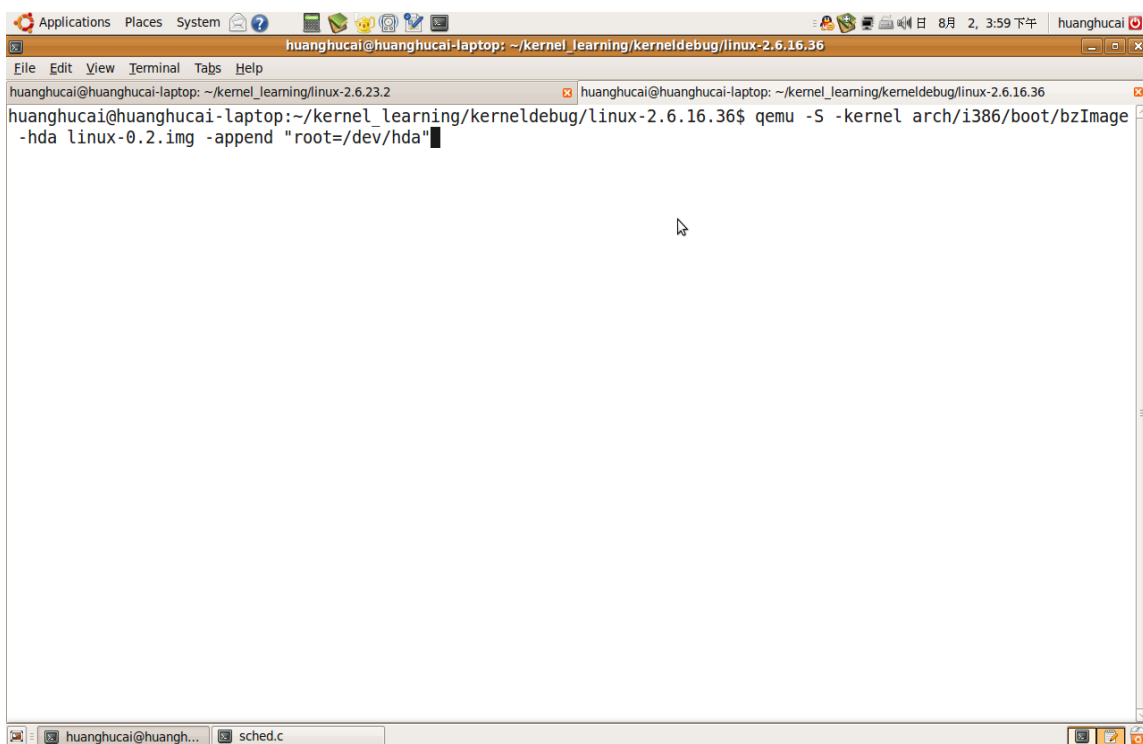
(三) QEMU 下内核源码级调试

首先是启动 qemu：

可以运行类似于下面命令来启动 qemu：

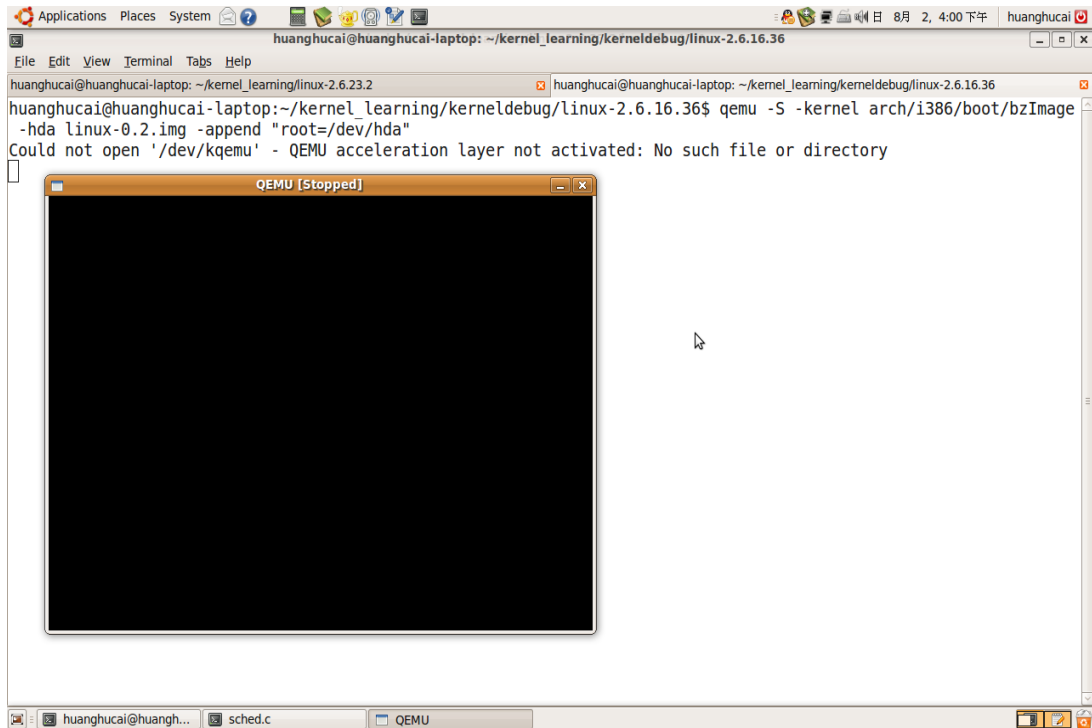
```
qemu -S -kernel arch/i386/boot/bzImage -hda linux-0.2.img -append "root = dev/hda" -no-kqemu
```

截图如下所示：



运行了此命令后就可以进行源码级的调试了。

当执行了上面的启动命令之后，会弹出下面的 QEMU 工作终端，

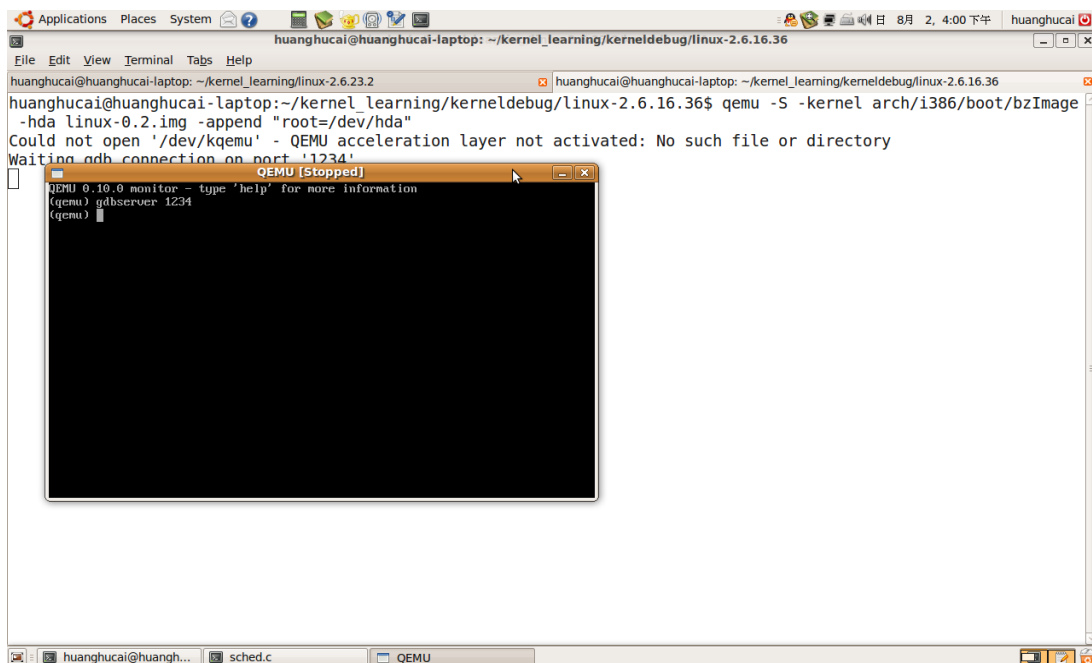


此时 QEMU 工作终端是一个黑色方框，没有任何内容，
现在要切换进 QEMU，按下 Ctrl+Alt+2 即切换到 QEMU 工作台下.启动 gdb 服务，并设置主机连接端口，使用下面命令即可实现：

`gdbserver 1234`

这里要理解的一点是:当启动 QEMU 后，QEMU 仿真器在等待着 gdb 的连接。

如图所示：

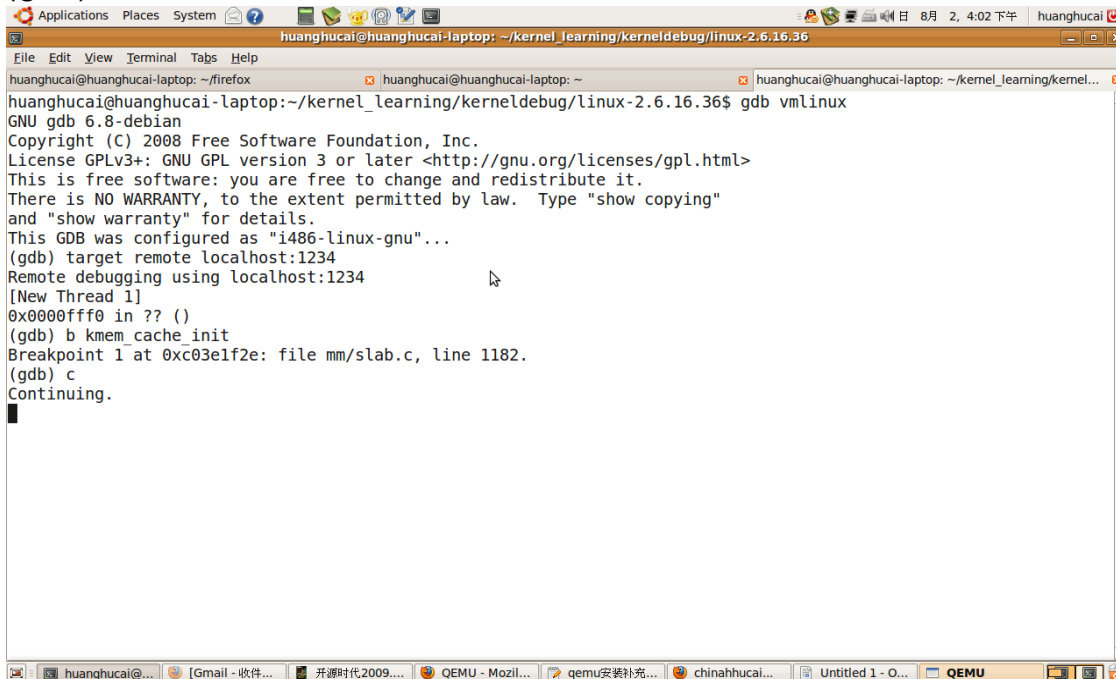


要在仿真器下进行源代码级的调试，需在另外一个终端下启动'vmlinux'调试，对应的命令为：
gdb vmlinux.这些工作都是在源码目录下实现的。

之后，在调试状态下进行 QEMU 的连接：

target remote localhost:1234

建立 QEMU 连接后，就可以进行 gdb 的正常使用了，例如，通过 c 命令启动内核
(gdb) c



```
huanghucai@huanghucai-laptop: ~/kernel_learning/kerneldebug/linux-2.6.16.36
File Edit View Terminal Tabs Help
huanghucai@huanghucai-laptop: ~/kernel_learning/kerneldebug/linux-2.6.16.36$ gdb vmlinux
GNU gdb 6.8-debian
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu"...
(gdb) target remote localhost:1234
Remote debugging using localhost:1234
[New Thread 1]
0x00000000 in ?? ()
(gdb) b kmem_cache_init
Breakpoint 1 at 0xc03e1f2e: file mm/slab.c, line 1182.
(gdb) c
Continuing.
```

Rootkit 技术的主要原理

ChinaUnix 网友: hackisle

rootkit 的主要分类：应用级->内核级->硬件级

早期的 rootkit 主要为应用级 rootkit，应用级 rootkit 主要通过替换 login、ps、ls、netstat 等系统工具，或修改 .rhosts 等系统配置文件等实现隐藏及后门；硬件级 rootkit 主要指 bios rootkit，可以在系统加载前获得控制权，通过向磁盘中写入文件，再由引导程序加载该文件重新获得控制权，也可以采用虚拟机技术，使整个操作系统运行在 rootkit 掌握之中；目前最常见的 rootkit 是内核级 rootkit。

内核级 rootkit 又可分为 lkm rootkit、非 lkm rootkit。lkm rootkit 主要基于 lkm 技术，通过系统提供的接口加载到内核空间，成为内核的一部分，进而通过 hook 系统调用等技术实现隐藏、后门功能。非 lkm rootkit 主要是指在系统不支持 lkm 机制时修改内核的一种方法，主要通过 /dev/mem、/dev/kmem 设备直接操作内存，从而对内核进行修改。

非 lkm rootkit 要实现对内核的修改，首先需要获得内核空间的内存，因此需要调用 kmalloc 分配内存，而 kmalloc 是内核空间的调用，无法在用户空间直接调用该函数，因此想到了通过 int 0x80 调用该函数的方法。先选择一个不常见的系统调用号，在 sys_call_table 中找到该项，通过写 /dev/mem 直接将其修改为 kmalloc 函数的地址，这样当我们在用户空间调用该系统调用时，就能通过 int 0x80 进入内核空间，执行 kmalloc 函数分配内存，并将分配好的内存地址由 eax 寄存器返

回，从而我们得到了一块属于内核地址空间的内存，接着将要 hack 的函数写入该内存，并再次修改系统调用表，就能实现 hook 系统调用的功能。

Rootkit 的常见功能：

隐藏文件：通过 strace ls 可以发现 ls 命令其实是通过 sys_getdents64 获得文件目录的，因此可以通过修改 sys_getdents64 系统调用或者更底层的 readdir 实现隐藏文件及目录，还有对 ext2 文件系统直接进行修改的方法，不过实现起来不够方便，也有一些具体的限制。

隐藏进程：隐藏进程的方法和隐藏文件类似，ps 命令是通过读取 /proc 文件系统下的进程目录获得进程信息的，只要能够隐藏 /proc 文件系统下的进程目录就可以达到隐藏进程的效果，即 hook sys_getdents64 和 readdir 等。

隐藏连接：netstat 命令是通过读取 /proc 文件系统下的 net/tcp 和 net/udp 文件获得当前连接信息，因此可以通过 hook sys_read 调用实现隐藏连接，也可以修改 tcp4_seq_show 和 udp4_seq_show 等函数实现。

隐藏模块：lsmod 命令主要是通过 sys_query_module 系统调用获得模块信息，可以通过 hook sys_query_module 系统调用隐藏模块，也可以通过将模块从内核模块链表中摘除从而达到隐藏效果。

嗅探工具：嗅探工具可以通过 libpcap 库直接访问链路层，截获数据包，也可以通过 linux 的 netfilter 框架在 IP 层的 hook 点上截获数据包。嗅探器要获得网络上的其他数据包需要将网卡设置为混杂模式，这是通过 ioctl 系统调用的 SIOCSIFFLAGS 命令实现的，查看网卡的当前模式是通过 SIOCGIFFLAGS 命令，因此可以通过 hook sys_ioctl 隐藏网卡的混杂模式。

密码记录：密码记录可以通过 hook sys_read 系统调用实现，比如通过判断当前运行的进程名或者当前终端是否关闭回显，可以获取用户的输入密码。hook sys_read 还可以实现 login 后门等其它功能。

日志擦除：传统的 unix 日志主要在 /var/log/messages, /var/log/lastlog, /var/run/utmp, /var/log/wtmp 下，可以通过编写相应的工具对日志文件进行修改，还可以将 HISTFILE 等环境变量设为 /dev/null 隐藏用户的一些操作信息。

内核后门：可以是本地的提权后门和网络的监听后门，本地的提权可以通过对内核模块发送定制命令实现，网络内核后门可以在 IP 层对进入主机的数据包进行监听，发现匹配的指定数据包后立刻启动回连进程。

Rootkit 的主要技术：

lkm 注射、模块摘除、拦截中断 (0x80、0x01)、劫持系统调用、运行时补丁、inline hook、端口反弹……

lkm 注射：也是一种隐藏内核模块的方法，通过感染系统的 lkm，在不影响原有功能的情况下将 rootkit 模块链接到系统 lkm 中，在模块运行时获得控制权，初始化后调用系统 lkm 的初始化函数，lkm 注射涉及到 elf 文件格式与模块加载机制。

模块摘除：主要是指将模块从模块链表中摘除从而隐藏模块的方法，最新加载的模块总是在模块链表的表头，因此可以在加载完 rootkit 模块后再加载一个清除模块将 rootkit 模块信息从链表中删除，再退出清除模块，新版本内核中也可以通过判断模块信息后直接 list_del。

拦截中断：主要通过 `sidt` 指令获得中断调用表的地址，进而获取中断处理程序的入口地址，修改对应的中断处理程序，如 `int 0x80`，`int 0x1` 等。其中拦截 `int 0x1` 是较新的技术，主要利用系统的调试机制，通过设置 `DR` 寄存器在要拦截的内存地址上下断点，从而在执行到指定指令时转入 `0x1` 中断的处理程序，通过修改 `0x1` 中断的处理程序即可实现想要的功能。

劫持系统调用：和拦截中断类似，但主要是对系统调用表进行修改，可以直接替换原系统调用表，也可以修改系统调用表的入口地址。在 2.4 内核之前，内核的系统调用表地址是导出的，因此可以直接对其进行修改。但在 2.6 内核之后，系统调用表的地址已经不再导出，需要对 `0x80` 中断处理程序进行分析从而获取系统调用表的地址。

运行时补丁：字符设备驱动程序和块设备驱动程序在加载时都会向系统注册一个 `Struct file_operations` 结构实现指定的 `read`、`write` 等操作，文件系统也是如此，通过修改文件系统的 `file_operations` 结构，可以实现新的 `read`、`write` 操作等。

inline hook：主要是指对内存中的内核函数直接修改，而不影响原先的功能，可以采用跳转的办法，也可以修改对下层函数的 `call offset` 实现。

端口反弹：主要是为了更好的突破防火墙的限制，可以在客户端上监听 80 端口，而在服务器端通过对客户端的 80 端口进行回连，伪装成一个访问 web 服务的正常进程从而突破防火墙的限制。

网友热评

热点技术评论

[8G 内存条，SWAP 分区要怎么划分啊？](#)

[有 root 权限,如何知道普通账户的口令？](#)

[crontab 不能执行 perl 脚本中的命令语句](#)

[Snort 入侵检测系统源码分析](#)

[lvm 的可靠性？](#)

[受到 ssh 攻击怎么办？](#)

[字符数组会自动扩容么？](#)

[同一 ip 上两个 socket 能否绑定到相同端口？](#)

[请教这个 shell 的写法，统计行数](#)

[多个 shell 程序，如何让其后台并发执行？](#)

[大家讨论一下 IM 的实现结构和方案](#)

[为什么这种写法不报错](#)

[TCP 连接中如何发出让对端不回复的数据包？](#)

[malloc 和 free 的困惑](#)

[问个 C++ 语法 类的成员函数声明而不定义](#)

[TCP-IP 详解卷 1 协议中关于 TCP 超时重传的问题](#)

[16 块硬盘存储数据库内容如何做 RAID？](#)

[文件夹批量逐级授权问题](#)

[问个问题：IBM 硬件分区](#)

[自定义函数和系统库函数冲突如何解决](#)

[交换两变量的值](#)

[x = x & \(x - 1\) 这个怎么理解呀？](#)

[文件系统这类的底层软件是不是一般用 C 编写](#)

[关于 gdb 调试的疑惑](#)

[你的代码平均每行值多少钱？](#)

热点新闻评论

[评论:雨林木风 Linux 意义和目的是什么？](#)

[作为一个 UBUNTU 用户，为什么要被鄙视？](#)

[母猪上树:微软首次向 Linux 社区提供驱动代码](#)

[ubuntu 真的很好吗？](#)

[Ubuntu 这点蛮恶心的！想解决啊！](#)

[请教一个最基础的问题---什么是"计算"](#)

[学校机房是否可以部分电脑不装操作系统？](#)

[韩国自主研发 16 年操作系统 Tmax 截图曝光](#)

[微软软件重大故障：伦敦证交所被迫停摆一天](#)

[现在感觉 ubuntu 就是桌面第一](#)

[Google 宣布将开发 ChromeOS 操作系统](#)

[卫星连接的革命 思科将把路由器送上太空](#)

[应届生找工作，这些你做到了么？](#)

[一位应届毕业生的经典简历](#)

[应届生找工作，这些你做到了么？](#)

[第二个番茄:雨林木风 Ylmf linux 1.0 版将发](#)

[你没见过的 helloworld 写法！](#)

[系统管理员的工作难吗？](#)

[请教一个面试题，关于大文件处理的](#)

[YLMF 雨林木风 Linux,垃圾](#)

[旗帜鲜明地支持雨林木风推出 linux 版本](#)

[程序员 30 岁以后怎么办？](#)

[CUer 浏览器使用情况调查](#)

[Windows 7 未上市 激活机制已惨遭破解](#)

[CentOS 玩完了么？](#)